

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
5 December 2002 (05.12.2002)

PCT

(10) International Publication Number  
**WO 02/097700 A2**

(51) International Patent Classification<sup>7</sup>: G06F 17/60

(71) Applicant (for all designated States except US): **THE CRAWFORD GROUP, INC.** [US/US]; 600 Corporate Park Drive, St. Louis, MO 63105 (US).

(21) International Application Number: PCT/US01/51431

(22) International Filing Date: 19 October 2001 (19.10.2001)

(72) Inventors; and

(25) Filing Language: English

(75) Inventors/Applicants (for US only): **WEINSTOCK, Timothy, Robert** [US/US]; 1845 Highcrest Drive, St. Charles, MO 63303 (US). **DE VALLANCE, Kimberly, Ann** [US/US]; 2037 Silent Spring Drive, Maryland Heights, MO 63043 (US). **HASELHORST, Randall, Allan** [US/US]; 1016 Scenic Oats Court, Imperial, MO 63052 (US). **KENNEDY, Craig, Stephen** [US/US]; 9129 Meadowglen Lane, St. Louis, MO 63126 (US). **SMITH, David, Gary** [US/US]; 10 Venice Place Court, Wildwood, MO 63040 (US). **TINGLE, William, T.** [US/US]; 17368 Hilltop Ridge Drive, Eureka, MO 63025 (US). **KLOPFENSTEIN, Anita, K.** [US/US]; 433 Schwarz Road, O'Fallon, IL 62269 (US).

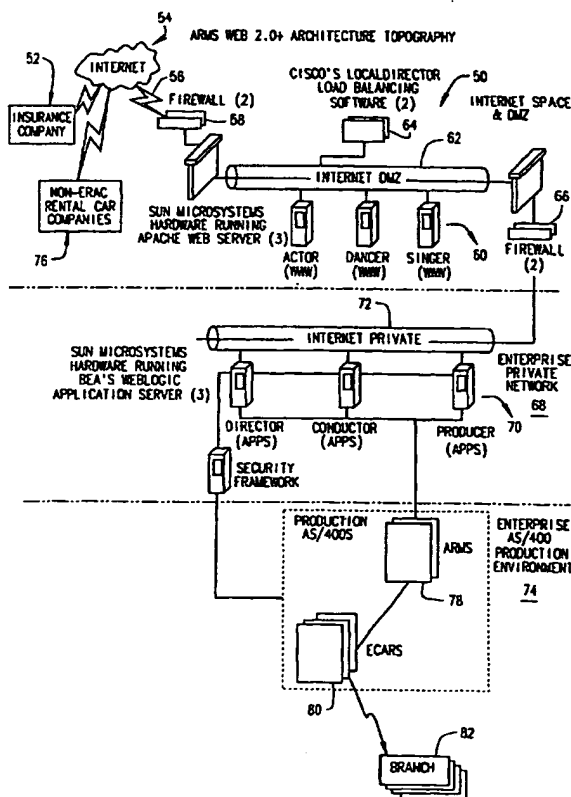
(26) Publication Language: English

(30) Priority Data:  
09/694,050 20 October 2000 (20.10.2000) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:  
US 09/694,050 (CIP)  
Filed on 20 October 2000 (20.10.2000)

[Continued on next page]

(54) Title: EXTENDED WEB ENABLED MULTI-FEATURED BUSINESS TO BUSINESS COMPUTER SYSTEM FOR RENTAL VEHICLE SERVICES



(57) Abstract: An Internet enabled, business-to-business computerized transaction system is disclosed in its preferred embodiment for use in providing rental car services for high volume users and comprises an Internet web portal through which the high volume user may access a plurality of service providers including an integrated business computer network for at least one rental vehicle service provider. The rental vehicle services provider computer network is configured to interconnect a geographically diverse plurality of branch offices, cataloguing their available rental vehicles and schedules for same as well as handling all transactional data relating to its business. The Internet web portal provides ubiquitous connectivity and portability for a multi-level business organization who regularly places high volumes of rental purchases with its business partner and also those other service providers who may or may not have the same integrated business computer system and software. Utilizing the method and apparatus of the present invention large volumes of rental transactions may be placed, monitored, altered during performance, and closed out with financial accounting and payment being made virtually without human intervention.

WO 02/097700 A2



(74) **Agents:** HAFERKAMP, Richard, E. et al.; Howell & Haferkamp, L.C., Suite 1400, 7733 Forsyth Blvd., St. Louis, MO 63105-1817 (US).

(81) **Designated States (national):** AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States (regional):** ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— *without international search report and to be republished upon receipt of that report*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

Extended Web Enabled Multi-Featured Business To Business  
Computer System For Rental Vehicle Services

Cross Reference to Related Application

This application is a continuation-in-part of Serial No. 09/694,050 filed October 20, 2000, which is a continuation-in-part of Serial No. 09/641,820, filed August 18, 2000.

5

Introduction

The invention disclosed and claimed in the first filed parent cross referenced above relates generally to the field of an Internet enabled business-to-business intelligent  
10 communication link allowing a first business organization to have intelligent interaction with a second fully integrated business organization to facilitate the placing of orders or reservations for business services or goods, with the services or goods provider having a computer network linking multiple  
15 levels of its organization to provide for the smooth conduct of business between the two organizations. More particularly, this field relates to an Internet enabled automatic rental



vehicle transaction system to facilitate the conduct of rental vehicle transactions between two multilevel business organizations, one of which provides such rental vehicle transaction services in an integrated manner through business enterprise software to a high volume user of such rental vehicle services wherein an Internet web portal is defined by the rental vehicle service provider which interconnects the two business organizations at multiple levels, providing a graphical user interface (GUI) for the transaction of large amounts of rental vehicle services automatically and virtually without human intervention upon entry. The invention of the second filed parent continuation-in-part application extends the functionality of the first filed parent invention by providing an intelligent portal that is readily configurable to suit any particular customer and any particular provider data requirements or method of doing business. This added functionality allows the invention, for example, to provide the user with access to other suppliers in the same seamless and integrated manner. In other words, the user now has access to not just one integrated business but multiple businesses, some of which may but need not be, integrated businesses thereby extending the invention for use in a generic application to satisfy a user's needs for a good or service not just from one vendor but all vendors connected to the invention. The inventions disclosed in this application add to the functionality of the systems first disclosed in the two parent applications by providing features and advantages which increases its flexibility and adaptability to other business models as might be found in different countries for handling rental vehicle transactions.

#### Background of the Invention

Computer technology has been embraced by many businesses in order to handle their ever increasing order flow as well as to mitigate the increasing blizzard of paper required to be

produced to document this business. A significant benefit which often drives the implementation of technology is its further advantage in increasing productivity to thereby allow fewer people to handle greater volumes of business. One such  
5 good example demonstrating the efficiencies and value to be gained by implementing technology is the business model developed and followed by the assignee of the present invention. A rental car company at its heart, the assignee transacts an ever increasing number of time sensitive,  
10 relatively low dollar volume, vehicle rentals which in many instances require authorizations to be made in advance, reservations of vehicles from available geographic and vehicle type selections, monitoring of the rental as it progresses including possibly extending the rental under certain  
15 circumstances, communications between the various parties involved in the transaction to ensure ultimate customer satisfaction, and financial accounting for the transaction including generating invoices and processing them for payment. While a significant portion of the vehicle rental business  
20 involves rental for leisure, business travel, etc., another significant business relationship has developed with insurance companies and the like in what has been termed as the replacement car rental service business. In this business, a vehicle insurance company may have many thousands of  
25 policyholders who are eligible to be involved in accidents, and other dislocations of use, requiring that a vehicle be rented for that customer's use while his own vehicle be made ready again for use. Thus, for this business segment, a multi-tiered business organization such as a vehicle insurance  
30 company represents a significant customer for repetitive vehicle rental services. To conduct this business in an orderly, time efficient and cost efficient manner, it is necessary that this insurance company has as its business partner a vehicle rental company which is itself multi-tiered,  
35 such as the assignee of the present invention. This is

because the needs, both geographically and in volume, are significant which require the dedication of a significant amount of resources. To satisfy these needs and to respond to other business growth, in its embrace of technology the assignee hereof has succeeded in developing an in-house computer system and related software which has integrated its business internally. This business integration has been massive and company-wide as is needed to integrate a company having a central office with literally thousands of individual branches located nationally, and even now internationally, with hundreds of thousands of vehicles available for rental. Furthermore, other business partners including other service providers such as vehicle repair shops have also been given access to this system to allow for input of information relating to progress of vehicle repair, extension of rental time, etc. as the rental progresses. This integrated business computer network and software generally includes a mainframe server at the heart of a wide area network (WAN) which facilitates the transfer of vehicle rental information and orders company-wide. This integrated business model is most efficient and needed in order to satisfy the vehicle rental service needs of a vehicle insurance company which itself may be national or even international in scope.

As a first step in extending the integration of technology into this business model, the present assignee has previously developed and implemented a computer system which has provided improved communication capabilities between the two business partners. This system generally comprised a second mainframe computer linked to the first mainframe of the integrated business network, with dedicated access lines being provided from this second mainframe to various levels of the multilevel business organization comprising the insurance company. In effect, with this additional mainframe and dedicated pipeline access, various individuals at the insurance company were permitted to directly interact with the

integrated business computer network of the vehicle rental company as well as other selected service providers such as body shops where wrecked vehicles were being repaired. The implementation of this system provided a great step forward  
5 over the people intensive business activity previously required in order to handle the large number of transactions encountered in this business relationship. Historically, the replacement car market engendered large numbers of telephone calls being placed between the insurance company, the rental  
10 company, and the body shop where vehicle repair was being performed in order to authorize the rental, select and secure the desired replacement vehicle to be provided, monitor the progress of the repair work so that scheduling of the rental vehicle could be controlled, extending the vehicle rental in  
15 the event of delays in repair, authorizing various activities involved in the rental process including upgrades of vehicles or other charges for services, and subsequent billing of the rental service and processing the billing to the insurance company for payment.

20 While the implementation of this system was successful and represented a tremendous step forward in automating the business relationship between the insurance company and the vehicle rental company, it did have certain limitations. For example, a specific communication link had to be established  
25 between the rental vehicle company and the particular users at the insurance company designated to have access to this system. Thus, special attention and some modicum of expense was required to establish these "pipelines" and maintain them. Still another aspect to the system implemented was that it was  
30 not "browser" based nor did it provide graphical user interface (GUI) menus. Thus, each user had to be specifically trained in the particular "language" used by the system and learn to work with specific menus nested in a specific manner as well as codes for entering commands which were not similar  
35 to other computer software programs. This software design

thus necessarily required additional training in order to insure that users could gain the full measure of advantage provided by the system and in order to minimize the opportunity for erroneous information or incorrect reservations from being entered or otherwise confusing the business transactions. Furthermore, user efficiency was not immediate and required skill beyond that ordinarily found in casual computer users, as we are all becoming in this computer age. Still another disadvantage to the system was that access was required to a designated entry point in the system in order for a person authorized to be on the system to work with it. As the nature of the insurance and replacement car business requires extreme mobility at multiple levels of both business partners, this represents a limitation to the usefulness and time efficiency with which various business functions could be performed. Therefore, while implementation of the second mainframe allowing for pipeline connections at various levels of the multi-tiered insurance company was a significant step forward in automating the business relationship between the two business partners, significant limitations to this solution were readily apparent to the users thereof.

#### Summary of the Invention

In the first parent application cross-referenced above, the inventors herein have previously succeeded in designing and developing a means for substantially enhancing the business to business communication link between these two businesses which provide significant advantages over its prior embodiment. More particularly, the inventors have succeeded in replacing the dedicated pipeline access of the existing system with a web portal allowing Internet access to the mainframe with a browser based graphical user interface (GUI) presentation. This also made the system more readily accessible to smaller business partners as the expense of the

"pipeline" was eliminated. The first parent's invention offers several important technical advantages over the previous system. First of all, by taking advantage of the ubiquitous nature of the Internet, the ultimate in portability and connectivity for this system is now provided in a business environment where mobility and connectivity are at a premium. In other words, a claims adjuster, body shop, or any other business employee authorized to have access to the system may gain access at any site offering Internet access. In present day technology that includes many mobile devices and appliances which are Internet enabled. As technology advances, it is conceivable that this access will extend to permit "24/7" access by any authorized person at any geographic location. This is a marked improvement providing immediate benefit and advantage over the dedicated pipeline access of the prior art system.

One limitation however, is that with this embodiment, this internet access must support a stateful connection. In this context, a stateful connection refers to a "persistent" conversation, meaning that the client side and server side software components establish a connection to one another once and multiple data transfers may occur without severing that connection. Common examples of a stateful connection include on-line chat, on-line gaming, and for virtually all on-line conferencing. This is distinguishable from the normal operation of web pages which typically establish a connection, transfer the object on the page, and then sever that connection. These types of connections are generally referred to as "stateless" connections.

A second major advantage of the first parent's invention is its graphical user interface. The inventors have taken full advantage of this browser based GUI to streamline and organize the presentation of information to a user to actually guide him as he interacts in doing his business. One such example is customized design of the menus such that the user

is guided and directed to answer only those questions required to be answered in order to conduct the particular transaction being addressed, and further to present choices to the user for his selection to minimize the need for the user to rely on his own memory or to be familiar with complicated and specialized codes to enter data or request transaction activity. With the recent and continuing explosion of the Internet, more people are becoming familiar with browser programs and their operation through their own daily activities in their personal lives. This familiarity paves the way for easier training and quicker orientation of a new user to the present invention. For large business organizations communicating at multiple levels, this significant advantage cannot be minimized as there are large numbers of people who must be continuously trained due to the growth of the organizations, as well as the replacement of employees due to the inevitable attrition. Thus, the first parent's invention provides an immediate increase in worker productivity, and makes that improved efficiency available to many more workers who are not particularly skilled otherwise in computer usage.

Still another advantage provided by the first parent's invention is through the implementation of additional functionalities which are engendered by the browser/GUI interface. As the system is continuously used, and feedback is continuously monitored and analyzed, additional features that add value through providing management information as well as by speeding transaction activity over the system may be implemented. For example, several of these features include the ability of a user to create an on demand report for transaction activity including summaries of transactions handled by a particular user or group of users which might either be open or closed. Another example of additional functionality which improves the efficiency of a user is the ability to create a repair facility call back list which

allows a user to sort existing open vehicle rental reservations by repair facility (body shop) and date such that a user is presented with the list of open reservations at a particular repair facility which can be readily handled in a single telephone call while at the same time having the system on line to implement any needed changes such as extensions of reservations, etc. Additional functionality has also been provided to speed the processing of invoicing which of course also speeds their payment and cash receipts. For example, it was found that even despite the built-in error checking and correction facilities provided to the users of the system, a repetitive pattern of mistakes involving incorrect claim numbers was discovered. To speed the processing of these, an additional functionality was provided as an "electronic audit" known as invoice return which returns an invoice to a particular adjuster upon detection of an incorrect claim number for his human intervention and correction of the claim number. In this manner, problem invoices exhibiting one of the most common problems encountered may be readily handled within the system and in an efficient manner, instead of manually as before.

The first parent's invention also has as a significant advantage the ability to be further customized to meet the individual business partners' needs and desires as well as to provide additional functionality by offering additional features which become desirable upon accumulation of user data based on user experience. Furthermore, once implemented, they are immediately available system wide. While this allows for consistent usage, it is limited in the sense that all of the system users are forced to use the same menus, data definitions, etc. This is not seen as a limitation for the one-to-one business application intended to be primarily addressed by the first parent's invention.

Still another advantage of the first parent's invention is that the graphical user interface incorporates point and



click interaction, using buttons and tabs to present or conceal data for the user's attention or inattention as the case may be, and provide a much more robust interaction capability through the creation of menu designs that allow for  
5 access to the most commonly needed features from any point in the menu architecture. This is to be contrasted with the prior system which consisted of a main frame character based interface while the first parent's invention with its GUI interface allows a user to point and click to navigate and to  
10 make selections by pull down selection, thereby reducing errors. As users become more experienced with the system, and their confidence level grows, they are much more likely to become bored and aggravated with the rigid structure of the prior system requiring them to follow along a certain menu  
15 architecture in order to complete certain tasks. On the other hand, the first parent's invention generally increases the interest of the user in using the system. These advantages of the first parent's invention over the prior interface promote employee productivity by allowing a user more control over his  
20 work which is critical in achieving savings in human resources to operate the system which is one of its main goals.

The second parent's invention extends the first parent's invention and expands its capabilities and functionalities. With the second parent's invention, a user may not only have  
25 access to its business partner, but also one or more competitors of its business partner through the same Internet portal. In this way, at least two needs are satisfied. First, the user can have access to a variety of providers to choose from where business needs or desires require. This  
30 allows the user to use a single portal and not have to sign on to a number of different portals, even should they be available. Furthermore, the user isn't troubled to learn how to access and use different portals even should they be available. Presently, not all providers are operating an  
35 Internet portal for offering their services, so by allowing

business competitors to be accessible through the same portal, independent development of other portals is forestalled. This is a benefit to the operator of the main portal as it creates and maintains a competitive advantage by handling all of the order flow which creates a data base of useful information for marketing purposes. Although initially the portal services might be offered for no additional cost to a competitor, eventually a fee might be charged which would at least partially offset the cost for owning and operating the portal.

10 The design of the portal is elegant and offers great flexibility for customizing not only the menus for presentation to the user, but also in the design of the data base entries needed or desired by the user and/or the competitive provider. For example, some users might not know or care about the features of a vehicle rented and so those data entries may not be provided space on the menu for the user to fill in. The data base as handled by the networked computer system then need not keep track of that data for that customer. This feature is readily accommodated by the data base programming and is conveniently implemented.

20 In still another aspect of the second parent's invention, the web portal has the capability to accommodate the varying data requirements also of the various competitive providers, but also the level of their sophistication as evidenced in their respective computer systems and interface facilities. For example, the web portal may be configured to communicate the user's order to the competitive provider via email, phone, or even through a connection directly to an integrated computer system having the same or substantially the same inter-operability as the integrated computer system of the assignee hereof. This capability extends to accommodating and matching the competing data requirements of the user and the competitive providers, and having the flexibility to design and implement menus that readily meet these competing needs. Furthermore, the second parent's invention allows for changes

to be implemented by simple re-programming of the web portal which minimizes the effort and enhances the "user friendly" aspect to the present invention.

Not only are these "global" improvements made available  
5 with the second parent's invention, there are other more particularized improvements that add functionality within the operating framework of the second parent's invention. For example, one such improvement is the ability to "virtually" assign work groups within the user so that, for example,  
10 multiple adjusters might be made into a team with a shared work load so that all of the team members have access to the same pool of work, such as the placing of reservations for the same group of drivers. With this "virtual team" assignment capability, work groups may be readily re-assigned to match  
15 changing work loads without worrying about re-configuring hardware or internal network connections. This can be a very valuable feature to accommodate staffing issues over geographical distances that can be nationwide, with access through the web portal to reservation facilities which are  
20 themselves nationwide.

Still another feature is the ability to customize an individual user's authorization limits. As can be appreciated, one of the mixed blessings of providing enhanced functionality to the individual users of any integrated  
25 computer system is that it places great power in the hands of the user which at the same time creates the potential for abuse. There have been well publicized instances of "rogue" employees making financial decisions or placing instructions which have far reaching financial consequences well beyond the  
30 intended authority of an employee, with disastrous results. With the second parent's invention, one feature is the ability to limit the financial commitments that a user may make during any pre-selected time period. For example, the user's profile may limit his ability to make only a certain dollar limit of  
35 vehicle reservations over any certain number of work days. In

this way, added safe guards may be conveniently provided, monitored by reporting capabilities, and changed as circumstances warrant, all with simple programming changes at the web portal.

5           There are still other features that are provided by the second parent's invention that find their genesis in the different approach taken over the first parent's invention and owing to the inherent increased flexibility of using a web based programming for the web portal to interface between the  
10 user and the providers on the web server and eliminating the need for any custom software on the user's terminal. The details of these are to be found and described in the detailed description of the preferred embodiment below. Examples include the ability to send confirmatory communications to the  
15 user that the reservation has been received and entered into the provider's system for fulfillment, custom report design including the capability to save and re-generate the custom report upon user command, increased flexibility to process and pay invoices, etc.

20           Still other advantages and features have been developed and are newly disclosed and claimed more particularly herein. These advantages and features relate to usage of the present invention both domestically and abroad where there are idiosyncrasies in the business model that need to be  
25 accommodated. Still other features provide entirely new functionality. One such new feature involves adapting the present invention as a tool to market replacement vehicles for sale or lease to a customer who has had an accident significant enough that repair of his vehicle is not  
30 economically feasible. This is commonly referred to "totaling" a vehicle. The insurance industry totals about 3 million cars per year, of which approximately 17% are newer models (defined as within three years of current model year). Once totaled, the owner needs to buy another car. Since car

rental companies desire to sell more cars, any opportunity to tap into the total loss market will be bountiful.

The present invention provides a window into the establishment of a total loss for a

5 renter's/insured's/claimant's automobile. Any car that is deemed to be a total loss would be indicated as such in the present invention for reporting purposes. At this point the stored information could be used to help provide economic benefit to all parties, insurance company, rental car company,  
10 and automobile owner.

Once a renter's/insured's/claimant's (owner's) car is determined to be a total loss the adjuster will try to ascertain the actual cash value (ACV) to be settled with the owner. The adjuster can use a third party tool, such as CCC's  
15 Pathways® product, to determine what ACV is. Today an adjuster must input this information manually into a separate application. The present invention contains much of the necessary information needed to determine ACV: name, car make, model series, year. The present invention need merely  
20 send the necessary information electronically to a total loss product and request an electronic response. Once the necessary information is generated, the present invention would in turn take the ACV and cross reference the car rental database of inventory.. Necessary information might include  
25 but not be limited to: ACV, year, make, model series, comparable cars, etc.

The car rental inventory can be filtered by geography and "holding requirements". As a reseller of vehicles, the car rental inventory is generally contractually required to be  
30 within the fleet as a rental for a predetermined amount of time prior to being available for sale to third parties. Once a car is past the holding requirement it is generally within the discretion of the car rental company to sell. Thus, instead of X% of cars available to the car rental company for

retail sale, a virtual inventory of cars is available for retail sale to the owner of the car.

Once the filters for geography and holding requirements are active, the present invention delivers a list of available  
5 vehicles for sale. At this point the adjuster and owner review the available cars, decide the cars considered to be attractive, and the owner then decides which one he wishes to purchase.

The user then selects one or more potential vehicles and  
10 sends the request to the appropriate car rental location. The car rental location can then contact the owner of the vehicle to buy one of the selected vehicles. In addition, the list of vehicles and ACV information can be sent to the owner for further review and discussion.

15 Once the car rental company contacts the owner and comes to a sufficient conclusion, either to buy or not to buy, the adjuster is notified of the conclusion and the transaction is consummated either through the present invention or off-line.

Still other features are disclosed and claimed herein  
20 which extend the functionality of the present invention. These include the following. One such feature is providing for automatic extensions of existing rental authorization, so that some limited extension authority is granted to permit some flexibility to a particular user without burdening him  
25 with the need to obtain approval for the extension. Another feature could be referred to offline usage, and provides the functional advantage of permitting processing of reservation data in a computer not connected into the network, and then uploading/downloading between the offline computer as it is  
30 connected into the network, such as by dialing into the network over the internet, or through a portal. The type of data which could be processed includes virtually any related to the processing of vehicle rental transactions and other related data such as car repair scheduling, etc. This  
35 functionality provides an extension of the usability to the

invention to mobile users who travel beyond the reach of the internet, which even further enhances its applicability to those places not covered by wireless coverage. Alternatively, it allows the invention to bypass special connectivity issues which are thought to be disadvantageous for any reason including cost, unavailability, inconvenience, etc. Still another feature includes further integration of the internal data bases kept by permitting a user to automatically update not just one but several data bases with a single command once that new data is entered into a single menu. For example, in what can be referred to as "power templates", a user may enter a multiple number of rental reservations on a single menu and then click a single "approved" icon which would then enter all of them into the system. This represents an improvement over a previous implementation requiring a user to separately "approve" each reservation, and then suffer the system processing time for each reservation. This "batch" processing can result in significant improvement in throughput, and reduction of user interface time for processing multiple transactions. Still another feature provides the added functionality of processing customer satisfaction feedback through the system. This feature provides the capability for a user to enter customer feedback information, both positive and negative but perhaps more importantly negative, so that immediate awareness of any problem can be obtained and corrective action taken to mitigate or eliminate the difficulty. This feature also allows a user to indicate a suggested supervisory level of interaction, or the system may allow for automatic escalation of involvement for succeeding levels of supervisory attention as the dissatisfaction continues or even escalates. This feature can be significant to a service provider as the ultimate success of a service provider is directly dependent on the perception of satisfaction by the end customer. And, it is well known that the sooner a problem is identified and solved, the more likely

a customer will have a satisfactory experience. Furthermore, from a strict economic viewpoint, the sooner some problem is addressed and solved, generally the less expensive the solution. A small accommodation can change a frown to a smile, if promptly offered.

Still other features are now disclosed that have applicability perhaps in the domestic business model, but certainly offer needed functionality in other business models found in other countries. One of these includes multiple party involvement/management of a rental transaction. While the flexibility of allowing multiple adjusters within a group to "work on" a rental transaction has been previously described, this particular feature is different in that not only may these multiple adjusters not be within the same group, they might not be employed by the same employer, might not be adjusters themselves, and might have different authority for action on the transaction as is commonly found in different countries. For example, in some countries one adjuster authorizes and manages the rental reservation for the car while another adjuster authorizes and manages the insurance coverage for the rental. Still another feature allows third party or "independent party" management of the rental. In some countries a third party other than an insurance company is involved, such as a "credit hire" or "assist companies" or "repair facility" or "lawyer" or "fleet management company". Each of these third parties, or any other third party, may be permitted access to the system and a user profile created for them that defines their authority to process rental transactions through an administrative profile set up in advance through agreement with the authorizing agent, such as an insurance company. As an enhancement, various individualized features may also provide data indigenous to a particular country, such as electronic access to the Schwackliste book for an adjuster to conveniently view a "class" for a car to determine what replacement vehicle is



legally authorized for rental. Still another example of a feature needed to accommodate international capability is a need for a tiered rate system, and an hourly rental charge instead of a daily charge which predominates the domestic market. Processing of electronic signatures to satisfy local custom or legal requirement is yet another example of a feature for which the present invention is uniquely suited to provide.

While the principal advantages and features of the invention have been discussed above, a greater understanding of the invention including a fuller description of its other advantages and features may be attained by referring to the drawings and the detailed description of the preferred embodiment which follow.

15

#### Brief Description of the Drawings

Figure 1 is a schematic diagram of the computer systems comprising the first parent's invention;

Figure 2 is a flow chart of the software programs which communicate over the computer systems of fig. 1 to implement the first parent's invention; and

Figure 3 is a schematic diagram of the computer systems comprising the second parent's invention.

25

#### Detailed Description of the Preferred Embodiment

The overall system architecture for the first parent's invention 20 is best shown in Fig. 1. As shown therein, an insurance company computer system 22, which itself may be virtually any computer configuration or even a stand alone PC accesses the Internet 24 through any convenient access point 26 such as even including an ISP (Internet service provider), as known in the art. Also connected to the Internet 24 is a web portal 28 which is preferably provided by a server appropriately programmed as explained herein below. This web portal 28 may be appropriately configured as desired to suit

any particular business relationship or arrangement, although preferably the inventors herein and assignee of this invention have determined that a 24/7 or full time connection to the Internet 24 is preferable, except for scheduled downtimes for maintenance, etc. The service provider 30 which for purposes of explaining the first parent's preferred embodiment is preferably a vehicle rental organization, has itself an Internet portal mainframe 32 connected by a bi-directional communication link 34 to a second computer network 36 which may itself preferably have a mainframe server 38. This second computer system 36 is preferably a network having a database 40 for communication with what may be thousands of branch offices each of which has its own computer interface 44 which communicates to this second mainframe server 38 to conduct the integrated business functions of a service provider organization. Instead of communicating with the branch offices directly, a reservation may be communicated to a centralized location for further processing, such as a call center, and then relayed on to an appropriate branch office. This might be desirable under certain circumstances, such as if a branch office is closed, or when a purchaser requires some specialized service such as close monitoring of the rental. This may be done electronically and automatically, or with human intervention.

It should be noted that the particular computer configuration chosen as the preferred embodiment of the first parent's invention may itself be subject to wide variation. Furthermore, the term "mainframe" as used herein refers solely to a computer which can provide large scale processing of large numbers of transactions in a timely enough manner to suit the particular business application. Preferably, as is presently used by the assignee hereof, an IBM AS/400 mainframe computer is used as each of computers 32, 38. However, as is well known in the art, computer technology is subject to rapid change and it is difficult if not impossible to predict how

these computer systems may evolve as technology advances in this art. For example, it is not beyond the realm of possibility that in the not so distant future a network of computers would provide the processing power to conduct these  
5 business operations as presently handled by "mainframe" computers. Thus, the term "mainframe" is not used in a limiting sense but merely to indicate that it is descriptive of a computer suited to handle the processing needs for a large scale business application.

10 It should also be noted that the communication link 46 extending between the server 42 and each of the branch offices 44 may have alternative configurations. For example, in some applications access over the Internet may itself be adequate, recognizing the vagaries of Internet service availability,  
15 reliability, and processing speed. Alternatively, this communication link 46 could well be a dedicated pipeline providing broadband service connection full time with back up connections to ensure continuous communication between a particular branch office or groups of branch offices and the  
20 service providers business operations computer system 36. Some branch offices might even be served through satellite links. Indeed, it is even possible that a mixture of these wide variations of service level be present within a single organization's structure depending upon communication link  
25 cost and availability balanced against service needs. It should merely be noted for present purposes that this communication link 46 serves as the electronic umbilical cord through which branch offices 44 communicate with the business computer system 36 of the invention.

30 Attached hereto as exhibits are functional descriptions of the software programs resident on the computers comprising the two computer systems 32, 38 which implement the first parent's invention. More particularly, attached hereto as Exhibit A is a functional description of the software to  
35 implement the integrated business functions resident on the

AS/400 or mainframe computer 38. Attached hereto as Exhibits B and C are related flow diagrams and explanatory text, respectively, for the software resident on the mainframe AS/400 computer 32. Attached hereto as Exhibit D is a functional description of the software resident on computer 32 but which also appears on the server 28 which creates the web portal for access to the mainframe 32 and its resident program. Server 28 may use a bi-directional GUI to character based interface translator program, well known to those skilled in the art, to present the displays and information obtained and transmitted between the user and the computer 32. However, the software of Exhibit D could also be run on server 28, as would be appreciated by those of skill in the art. It is believed that these functional descriptions and accompanying text as exemplified in these exhibits are adequate to enable an ordinary programmer to implement corresponding software programs for executing the preferred embodiment of the first parent's invention using ordinary programming skills and without inventive effort.

As a further example of the flow of data and the functional advantages provided by the first parent's invention, reference is made to Fig. 2. As shown therein, a right hand column is identified as "ECARS" which represents the integrated business software implemented as part of the mainframe operation 38 in computer network 36. The center column headed "ARMS" is resident on mainframe computer 32 and coordinates the communication of data. The left column headed "ARMS/WEB" represents the software resident on computer but which is presented on server 28 and accessible by users through the Internet. Along the left side of Fig. 2 are designated three separate sections of operational activity. These are "reservation" followed by "open" and concluded by "close". Generally, the functional descriptions are arranged in chronological order proceeding from the top of Fig. 2 to the bottom. However, some functional features are permitted

throughout the entirety of one of the three periods designated at the left side of Fig. 2. One such example is the "message" function which allows messages to be sent between users at one business organization 22 and branch offices 44 and others  
5 connected to the other business organization 30. Proceeding with a description of the transaction, the first set of communications allow for the reservation of the services. These can include requests for authorization or a rescind authorization request to be sent from the service provider to  
10 the service purchaser. Correspondingly, authorizations and authorization cancels can be sent from the services purchaser to the services provider. Confirmations are communicated upon confirmation of an authorized reservation request. Authorization changes may be made and communicated from the  
15 services purchaser to the service provider. Corresponding rental transaction changes may be communicated from the services provider to the services purchaser. As indicated, through the entirety of this process messages may be sent between users and others connected or having access to the  
20 integrated business software, as desired. The consummation of this portion of the transaction is a reservation that has been placed, authorized, confirmed, and provision is made for changes as necessary. During the next phase of the transaction, a reservation is opened and services intended to  
25 be provided are started. Generally, and preferably for the rental of vehicles, a start and end date are established in the reservation process. However, along the way, transactional changes may be made, such as for changing the type of vehicle provided, extensions may be requested and  
30 entered from either business partner, messages may be transmitted between the business partners, and the transaction may be terminated such as by voiding the contract by one business partner or terminating the authority by the other business partner. The term "reservation" has been used herein  
35 to refer not only to the act of placing the order but also to

filling the order for services including providing the rental vehicle to the ultimate user and even invoicing for those services.

5       The last phase of the process involves closing the transaction. During this phase of the transaction, the contract is indicated as being closed and invoiced, the services purchaser can approve invoices, reject invoices, and also remit invoices. Such invoice remittance may also include the actual transfer of funds through an electronic funds  
10 transfer medium, or otherwise as previously arranged between the business partners.

      It should be understood that this is a streamlined description of the handling of a transaction, and by no means is exhaustive. For example, much more functionality is  
15 available to the user including accessing the data base to generate production reports regarding status of open or closed reservations, preparing action item lists to allow a user to organize and prioritize his work, obtaining information available in the system from having been entered by others  
20 which would otherwise require phone conversations which are inefficient and occupy still another person's time. A more detailed explanation of the functionality provided is found in the exhibits.

      In summary, the first parent's invention creates almost  
25 an illusion that the services purchaser, and the great number of users at various levels of the multi-tier purchaser users, are actually part of the services provider organization in that immediate online access is provided to significant data which enable the user to make reservations for services,  
30 monitor those services as they are being provided, communicate with those providing the services, obtain information relating to the status of services as they are being provided, and close transactions, all by interacting with the services provider business organization over that user's PC and without  
35 human interaction required by the business providers

personnel. By way of contra-distinction, for many years business has been conducted on a human level by customers picking up the telephone and calling services providers and talking to their human counterparts in order to convey  
5 information, place orders, monitor orders, including obtaining information as to status, canceling orders, questioning invoices and paying invoices, along with a myriad of other related interactions. Not only did the conduct of business in this manner entail significant amounts of human resources at  
10 both ends of the transaction, but it also led to inefficiencies, mistakes and delays all of which increase the cost of doing business and contribute to an increased risk of services being rendered in an unsatisfactory manner in many instances to the end user. The first parent's invention has  
15 taken the preexisting solution of providing electronic communication between the business partners to another level by "web enabling" this system for improved connectivity, improved usability, reduced training, enhanced mobility, and other advantages as described herein.

20 A schematic diagram of the second parent's invention is shown in Fig. 3 and includes three levels of architecture. As shown in the first level of the architecture 50, a user 52 such as an insurance company or other user has access through the Internet 54 to the computer system comprising and  
25 incorporating the invention. An Internet provider provides a link 56 through which Internet connections may be made to communicate with the further described system. For convenience, this Internet connection may be considered as an Internet site or portal in that a user enters a URL and  
30 arrives at this connection. A firewall 58 as is known in the art is used for security purposes and to prevent hackers and the like from unauthorized access to the system. A first set of servers 60 are interconnected in a network 62 and may preferably include an ancillary server 64 for running load  
35 balancing software or the like to balance the load and provide

redundancy amongst what may be a plurality of web servers 60. These web servers 60 may preferably be Sun Microsystem servers running Apache web server software, or other such suitable software as would be well known to those of ordinary skill in the art. This first web server network of servers 60, 62 process the random and disorderly communications flowing to and from this system and the Internet before passing them through a firewall 66 as a further precautionary measure. This first layer of architecture, identified as the Internet space/DMZ layer provides a secure interface and creates order out of the chaos of communications flowing between the system and others, as will be described.

With this architecture, stateless connections are accommodated, for the first time. By supporting stateless connections, this embodiment eliminates the implementation difficulties encountered with the first parent's embodiment on the client. These implementation difficulties include installing extra software on the client side computers, and eliminates the need for special configuration of the internet access method, such as proxy servers or routers. For example, many proxy server are configured to disallow stateful connections for security reasons, i.e. to prevent unauthorized programs from establishing such connections. Another example is that routers are customarily configured with most ports closed and thereby unable to support stateful connections.

The next layer of architecture 68 is noted in the figure as the "Enterprise private network" and is comprised of a plurality of servers 70 network connected with a network connection 72. Again, although the choice of hardware is not considered critical by the inventors hereof, Sun Microsystem's server/work station hardware is preferably used to provide the platform for running the application software for processing the various rental vehicle transactions, as will now be explained. Attached hereto as Exhibit E are a series of functional design specifications for the ARMS/WEB application



software resident on servers 70 and which provide the detailed description of the operational features of the software and system. With these functional design specifications for the individual modules, it would be readily apparent to those of  
5 ordinary skill in the art that programmers of ordinary skill would be able to write software to execute these functional specifications without using inventive effort. Furthermore, the details of this implementation are not considered to provide any aspect of the best mode for carrying out the  
10 invention which is defined by the claims below. Generally, the ARMS/WEB application software permits a user to sign on and, when recognized, provides the series of menus presenting choices for the user to indicate the parameters for his reservation. A plethora of information is provided and  
15 accessible to the user through the various menus provided from which the user selects and enters data to process the reservation. An important feature of the ARMS/WEB application software is that it provides the user the opportunity to select to place his vehicle rental reservation not only with  
20 the integrated business computer system represented by the third level of architecture 74, described below, but also to route the reservation information back through the first architectural level 50 and into the Internet 54 for transmission to a competitive service provider 76. Although  
25 the interconnection is depicted in Fig. 3 as being made through the Internet 54, the network of servers 70 configured in accordance with the ARMS/WEB application software may utilize virtually any electronic means for transmitting the reservation information to a competitive services provider 76.  
30 These include email, automated telephone, facsimile, and other forms of electronic communication. Of course, the competitive services provider 76 may itself comprise an integrated business such that the level of interconnectivity provided to the user 52 may parallel that disclosed and described in  
35 connection with the integrated services provider system of the

invention as well as the first parent's invention. This integrated business capability is represented as the third level 74 of the architectural topography shown in Fig. 3 which parallels portions of that shown in Fig. 1 in that a pair of  
5 network mainframe computers, such as AS/400's 78, 80 may process reservations to and from various branch offices 82 which are geographically diverse.

With the invention, the Internet portal provided by the ARMS/WEB network configured servers 70 provide an Internet  
10 portal for communication with not only the integrated computer enabled business system of the resident services provider, but also a portal for placing reservations to other competitive services providers 76. Thus, the user 52 enjoys the capability of accessing multiple service providers for  
15 competitive services through a single Internet connection using a single set of protocols, menus, etc. for the conduct of this business activity. Furthermore, the software configured network of servers 70 is readily configured in Web Logic to adapt to changing user requirements, data  
20 requirements, unique competitive service provider requirements, and other upgrades or modifications in a convenient manner by simply modifying the software resident therein. No special browser software or other interface software is required by the user and any special  
25 interconnecting software or server/hardware requirements may be satisfied as between the service providers such that the user is presented with a seamless interconnection. As the invention is configured and works well with the integrated business and computer systems as disclosed herein, it is  
30 anticipated that such interconnection and usability may be readily translated to any other such integrated computer system as might be found in other competitive service providers, as would be apparent to those of ordinary skill in the art. Thus, with the invention, a user is provided with  
35 among other things Internet access through a single portal to

a plurality of service providers and, to the extent possible, to their integrated computer business systems.

The invention is sufficiently flexible to accommodate changes which are intended to adapt it for use with other  
5 business models, and especially those encountered in other countries. Furthermore, some of these changes add features that are equally applicable domestically. One such example is an "automated extensions" feature. Typically, there are many occasions when a damaged or inconvenienced vehicle is not made  
10 available for use when originally scheduled. In the prior art, many times an extension would then need to be requested through the system, with authorization requested and provided. In order to streamline this process, and to minimize delay and involvement of supervisory authority, the system may provide  
15 for some form of automatic extension authority. Preferably, this could be provided in any one of three modalities, or some combination thereof. A first modality would be for the service provider to have automatic extension authority, upon communication to the customer, within certain pre-determined  
20 limits. For example, an initial authorization may be for 12 days of a vehicle rental. A request for an extension of 5 days may be made by the service provider and of that 5 days 3 days may be authorized automatically as being within 25% of the original rental term and a request for the additional 2  
25 days requiring approval may be automatically generated. Still another variation would be for the insurance company to set a limit within the system of the total number of authorized days, which could be based on some other parameter such as labor hours or body shop hours or down time for the repairs to  
30 take place. Then, upon request for an extension, one may be automatically granted based on the total authority allowed or initially set into the system by the insurance company, and up to that limit. Still another variation would be for a third party service provider to be involved in the process, such as  
35 a body shop, to make direct input into the system of a need

for an extension. These authorized third party providers would preferably be pre-selected and their authority limited as described above. This feature may be implemented conveniently in a separate menu, for example as shown in the  
5 attached "screen shots" headed "Extend Rental".

Another feature is an offline usage feature which allows a user, such as an adjuster, to work with a laptop having loaded thereon a software program that emulates the connected network software for local processing of data, such as claims  
10 data. In use, an adjuster would preferably first connect to the system and download or "synch" his laptop data base with the claims data resident in the system. The adjuster would then disconnect and use his local program to work offline. Such work could include the generation of new reservations,  
15 authorization of direct billings, extension of rentals, approval of invoices, and setting of termination dates for on-going rentals, among other tasks. The user would then re-connect to the system, such as over an internet connection, sign in, and "synch" his laptop to the system which then  
20 transmits or executes his commands/communications to the central processor. The central processor checks the users "synch" data against its data file, advises the user of any "synch" data that is older than the current data, and requests the user to specify which data should be processed. After the  
25 processor is instructed by the user, it will then act on the "synch" data. For clarity, a first "screen shot" is provided that illustrates a sign in log for a user who wants to initiate a "synch", and a second "screen shot" is provided to illustrate a listing of activity that could have been created  
30 offline and which is available to be input to the system upon "synching". A preferences feature is provided to allow a user to establish defaults for automatic syncing of the data. Also, a history feature will allow the user to display all of the syncing activity from his connection or portal including  
35 error messages and conflicts noted.

Yet another feature allows for a user to enter, or execute, a full menu of transactions without individually opening them from a summary menu. This has been referred to as a "power template" feature. Instead, a hyperlink is  
5 provided to allow a user to jump into another menu of details for an individual item should it need to be changed and not entered as suggested, requested or listed on a user's action list.

Still another feature allows for the collection of user  
10 satisfaction feedback, and alerts to be entered for the attention to complaints, by the user right at his terminal. This capability allows for a text message to be entered as well as the name and contact information of the party making the feedback. As known in the service industry, and as  
15 discussed above, customer satisfaction is important and the faster a complaint can be registered and communicated to the proper person for correction, and then corrected, the more likely that a customer will view his experience favorably. By providing a pop up menu item capability, a user may from any  
20 one of a number of menus immediately enter the description of the problem and send it to the proper person electronically with a minimal amount of effort and a high degree of reliability. A convenient record may then be made of these "feedback" issues and entered into the system database. With  
25 this information stored electronically, it may be conveniently searched and analyzed for any recurring patterns, thereby identifying any particular person, branch, facility, or type of problem that should be addressed for action beyond the solution of the immediate problem. A "screen shot" is  
30 provided to illustrate how the "pop up" menu may appear, although it could be varied to allow for entry of other or additional information such as "trouble codes" allowing for the type of problem to be user classified, etc. A flow diagram is also provided to illustrate the flow for  
35 complaints, a methodology for processing them including

escalating their importance and level of attention as the matter remains unresolved over time.

Still another feature that adds to the flexibility of the invention is a multiple adjuster feature, that can be extended  
5 to include an independent party control feature. In some countries, and in some business models either domestically or abroad, it may be preferable to have more than one adjuster be empowered to interact with or authorize certain facets of a vehicle rental transaction. In those situations, the  
10 invention can provide the flexibility and control needed to separately empower and control the interaction of multiple adjusters. For each user of the invention, an "Administration" schedule is set up by an authorizing agent, such as someone at the supervisory level of either the  
15 insurance company or the service provider, which grants authority for performing certain work activities as well as possibly limiting the amount of monetary authority allowed that adjuster. A "screen shot" is attached which exemplifies such authorization, with work activities including  
20 creating/authorizing reservations, maintain/extend rentals, pay invoices, user maintenance, receive unassigned action items, and reporting. This capability could be used to separately authorize different adjusters acting on behalf of the insurance company and the individual. In other words, the  
25 individual may need the car for 5 days but the individual's insurance coverage may only apply for 3 days while the insurance may pay for five days rental. This capability may also be further extended to independent third parties.

As extended for independent party management, this  
30 capability further adapts the invention for use with agencies such as "credit hire", "lawyer", "fleet management companies", or "repair facility", or "assist companies", all of which are found in other than domestic markets. Included herewith is an attachment which further explains the different types of  
35 independent parties routinely found at present, and examples

of "screen shots" which provide the additional functionality of customizing authorizations for each of these independent parties for interacting with a rental transaction.

Yet another feature provided by the invention is a  
5 facility for marketing cars for sale/lease to customers. As explained above, a customer will occasionally be forced to replace his vehicle at the same time that he is renting a vehicle for temporary use. Furthermore, the value of the replacement vehicle, or the approved value that an insurance  
10 company will allow under coverage, many times determines the available vehicles from which a customer will be allowed to select without personal expense. The invention is uniquely designed to provide a listing of available cars, and information about the cars, all from the existing rental car  
15 data base as is kept in routinely running the rental car company's main business of renting cars. It is a simple matter to provide a menu which allows a user to specify search through the car inventory with parameters such as zip code, vehicle category, make and model. Using any one or more of  
20 these parameters, a search inquiry will then produce a listing of available vehicles matching the parameters, along with additional information about the vehicle including mileage, selling price, and color as well as other accessories. A customer could then be advised of the search results and  
25 allowed to select a vehicle. The invention may, if agreed to by the insurance company, and possibly conditioned on the physical inspection of the car by the customer, then authorize the transfer of the vehicle to the customer as an outright settlement of his claim.

30 In implementing the replacement of the customers vehicle, a process preferably comprises the steps of an adjuster identifying the loss as a total loss which is preferably entered at the same time that a replacement vehicle rental is reserved, sending the vehicle data to a third party valuation  
35 tool for processing, determining the valuation of the vehicle

by a suitable measure such as actual cash value (ACV), sending the ACV to the system, using the search function to identify possible replacement vehicles available for the customer, finalizing the replacement process with the customer including  
5 executing transfer of title documentation if desired, and posting the results of the vehicle replacement in the system for access by the insurance adjuster so that he can confirm that the customer's claim has been satisfied. A flow chart describing this process is attached for further explanation.

10 Various changes and modifications to the preferred embodiment as explained herein would be envisioned by those of skill in the art. Examples of these changes and modifications include the utilization of computer systems configured in any one of a myriad of ways using present technology alone. For  
15 example, mobile computers are presently available and wireless technology could be used to extend the integrated business network of the services provider, as well as match the mobility needed by the various users connected to and using the present invention. The particular software, and various  
20 aspects and features of its design, have been adapted for particular application to the vehicle rental business. Of course, computer software applications satisfying other business needs would necessarily require adaptation to their particular business models. Thus, it is envisioned by the  
25 inventors herein that the various software programs described herein would be matched to the particular business application to which the invention is utilized. These and other aspects of the preferred embodiment should not be viewed as limiting and instead be considered merely as illustrative of an example  
30 of the practical implementation of the present invention. These changes and modifications should be considered as part of the invention and the invention should be considered as limited only by the scope of the claims appended hereto and their legal equivalents.



# **ARMS Web User Guide**

---

## **Automated Rental Management System**

### **Table of Contents**

<b>Introduction .....</b>	<b>i-v</b>
<b>Create Reservation .....</b>	<b>1-5</b>
<b>Change Reservation/Rental .....</b>	<b>1-12</b>
<b>Enterprise Requests-by Adjuster .....</b>	<b>1-8</b>
<b>Enterprise Requests-Unassigned .....</b>	<b>1-5</b>
<b>View Reservation/Rental .....</b>	<b>1-6</b>
<b>On-Line Reporting .....</b>	<b>1-6</b>
<b>Process Approved Invoices .....</b>	<b>1-7</b>
<b>Setup and Maintenance .....</b>	<b>1-7</b>
<b>Buttons .....</b>	<b>Appendix</b>

# Introduction

---

## Introduction

<b>Welcome to ARMS Web! .....</b>	<b>i</b>
<b>What is ARMS Web? .....</b>	<b>i</b>
<b>User Guide Overview .....</b>	<b>ii</b>
<b>Helpful Hints .....</b>	<b>ii</b>
<b>Common Definitions .....</b>	<b>ii</b>
<b>Function Keys .....</b>	<b>iii</b>
<b>Other Information .....</b>	<b>iii</b>
<b>On-screen Information.....</b>	<b>iii</b>
<b>Sign On .....</b>	<b>iv</b>
<b>ARMS Web Main Menu .....</b>	<b>v</b>

## **Introduction**

### **Welcome to ARMS Web!**

This User Guide offers step-by-step instructions that clearly guide you through each function of ARMS Web. It also shows you how to maximize the benefits of the system. Enterprise encourages you to keep the User Guide handy for future reference.

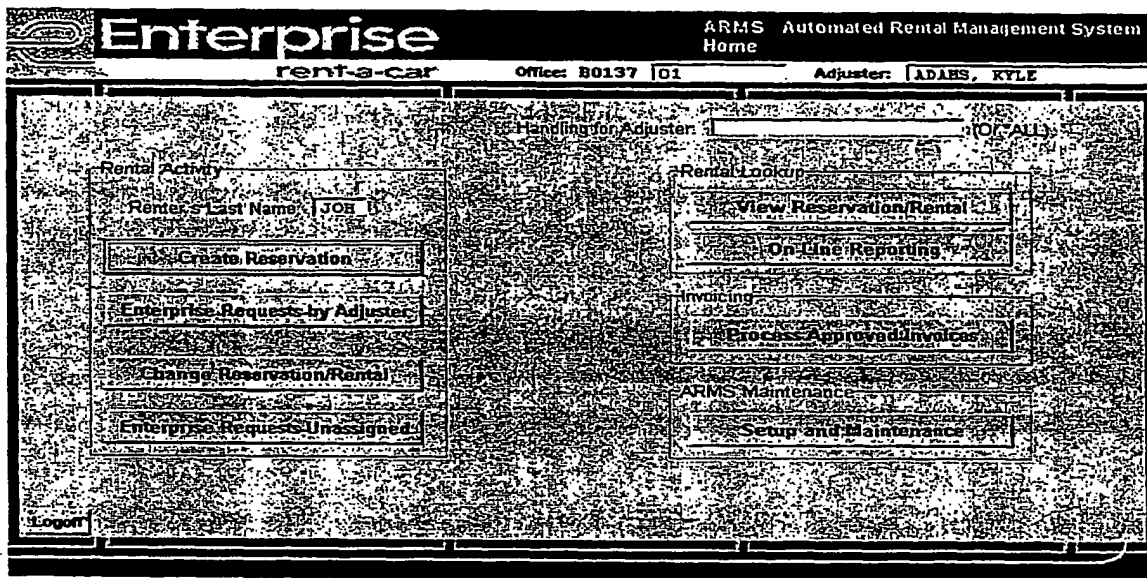
### **What is ARMS Web?**

It is the Internet-based version of the Automated Rental Management System. ARMS Web forms a web-based computer link between your company and Enterprise Rent-A-Car. You can create reservations to set up rentals for your insured or claimants directly through the Enterprise computer system. You also have the ability to change existing reservations and rentals, view rates, extend authorizations, receive authorization requests and extension requests, send messages back and forth with Enterprise, and much more.

## User Guide Overview

Keep these important points in mind when reading your User Guide:

The screens shown throughout this guide are reproduced directly from the ARMS Web system and are accurate reflections of the actual screens. Field buttons appear in bold type. They can be clicked to access information or move to another screen.



### Helpful Hints

- ◆ Remember to key your **claims office code** in the **Office** field on the main menu (shown above). Your name will already be set in the **Adjuster** field after you log on to the system. Once you enter the office code here, it is carried forward throughout the program for you.
- ◆ If you are monitoring or making changes to another adjuster's files, be sure to enter his or her last name in the **Handling for Adjuster** field.
- ◆ Whenever a button is highlighted (see Create Reservation above), you can press **ENTER** or click the button to select it. You can highlight a button by pressing **TAB** to move from one button to the next. Though there are three methods by which to select the screen, the User Guide will use only the highlight and click **OK** method. Remember that you can use any of the methods at any time.

### Common Definitions

- ◆ A *finger* is what the cursor turns into when it is positioned over a button. Click to select that button.



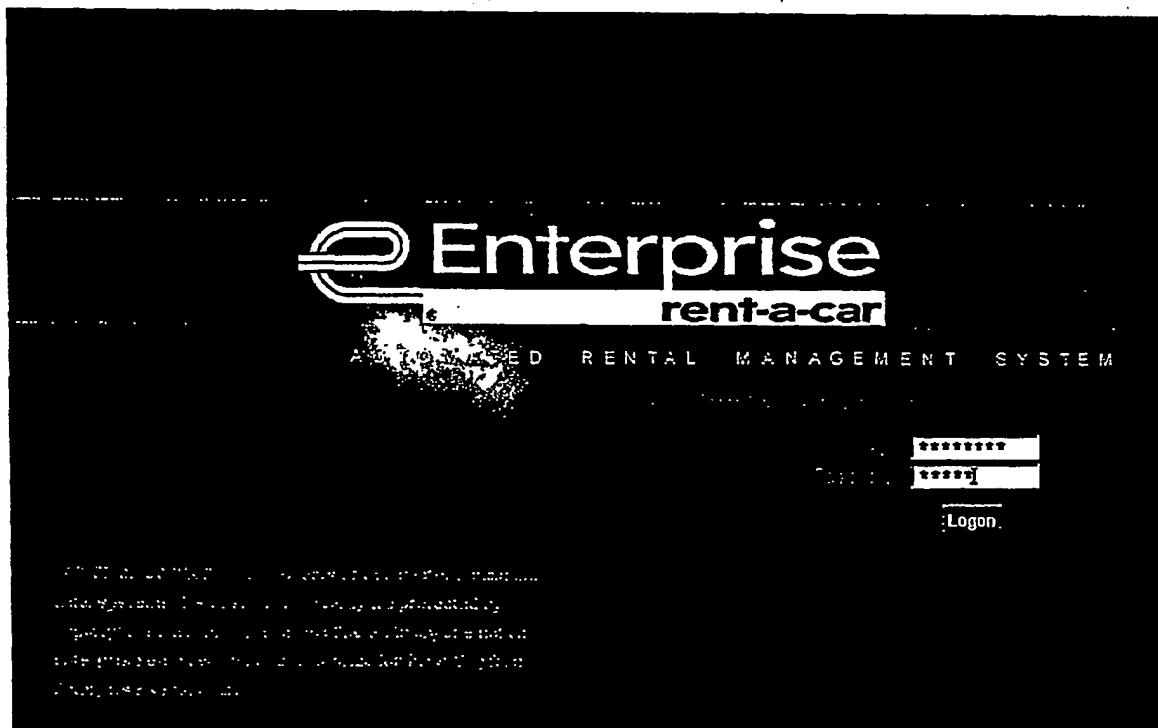
- ◆ A *button* is an area on the screen that allows you to access, view, and/or change information.

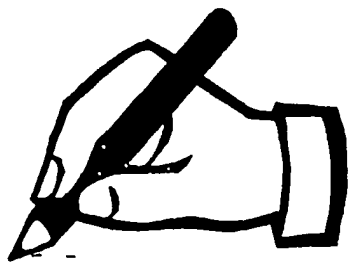
## Sign On

Depending on the type of link your company has with Enterprise, the following screen may or may not appear. If it does *not* appear, consult your supervisor for sign-on procedures.

If the Sign On screen *does* appear,

- 1) Key your User ID and Password. This user ID and password will be issued to you by Enterprise. These allow Enterprise to confirm that you are authorized to use ARMS Web.
- 2) Press **ENTER** or click Logon. The ARMS Web main menu appears (page v).





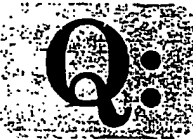
# notes



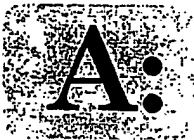
# Create Reservation

---

Create Reservation



When should I use Create Reservation?



Use this section to authorize a rental.

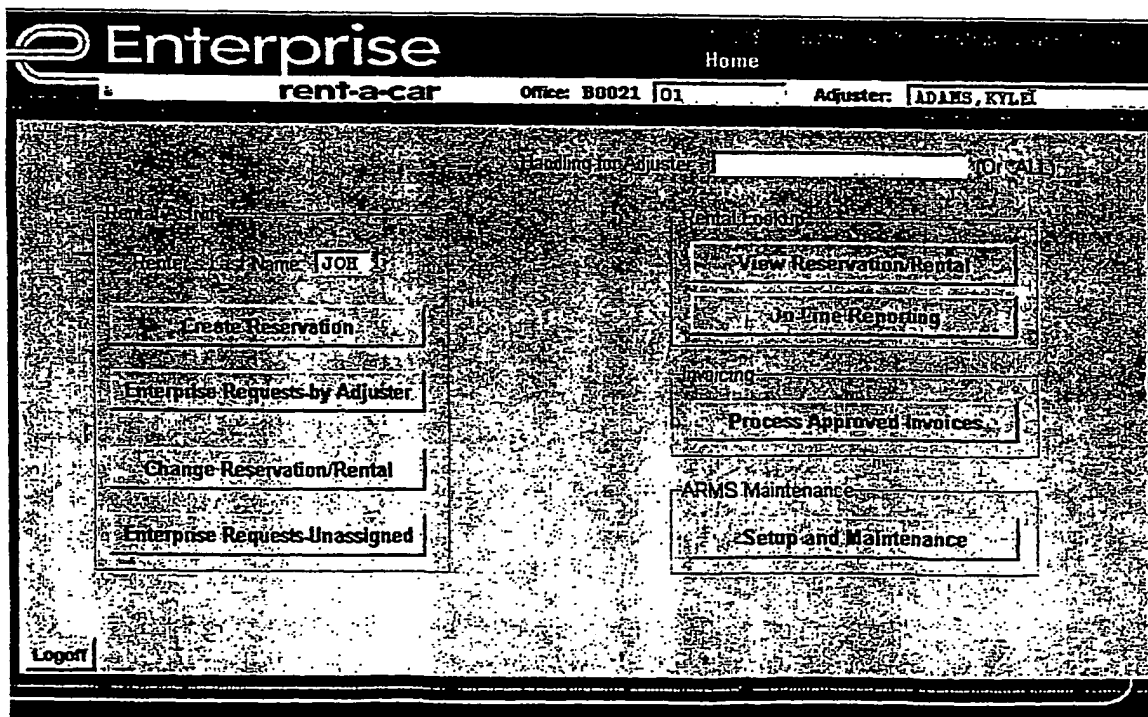
Create a Reservation - Main Menu .....	1
Select a Reservation .....	2
Select an Adjustor .....	3
Create a Reservation .....	4
Create a Reservation - Rate Selection .....	5

## Create a Reservation - Main Menu

- 1) Key your **claims office code** in the **Office** field.
- 2) Key your **last name** in the **Adjuster** field.
- 3) Key the first three letters of the **renter's last name** (required).
- 4) Click **Create Reservation**.
- 5) If Enterprise has already requested a reservation authorization for a renter whose last name matches the three letters keyed, the **Select Reservations/Rentals** screen appears when you click **Create Reservation** (page 2). If there is no match for the letters keyed, the **Create Reservation** screen appears (page 4).



The Select Reservations/Rentals screen prevents duplicate authorization requests. You will probably see it often.





## Select an Adjustor

This Adjuster Selection pop-up window appears when no adjuster has been assigned to the selected claim.

- 1) Key the desired adjuster name in the Adjuster Last Name field and click Search.

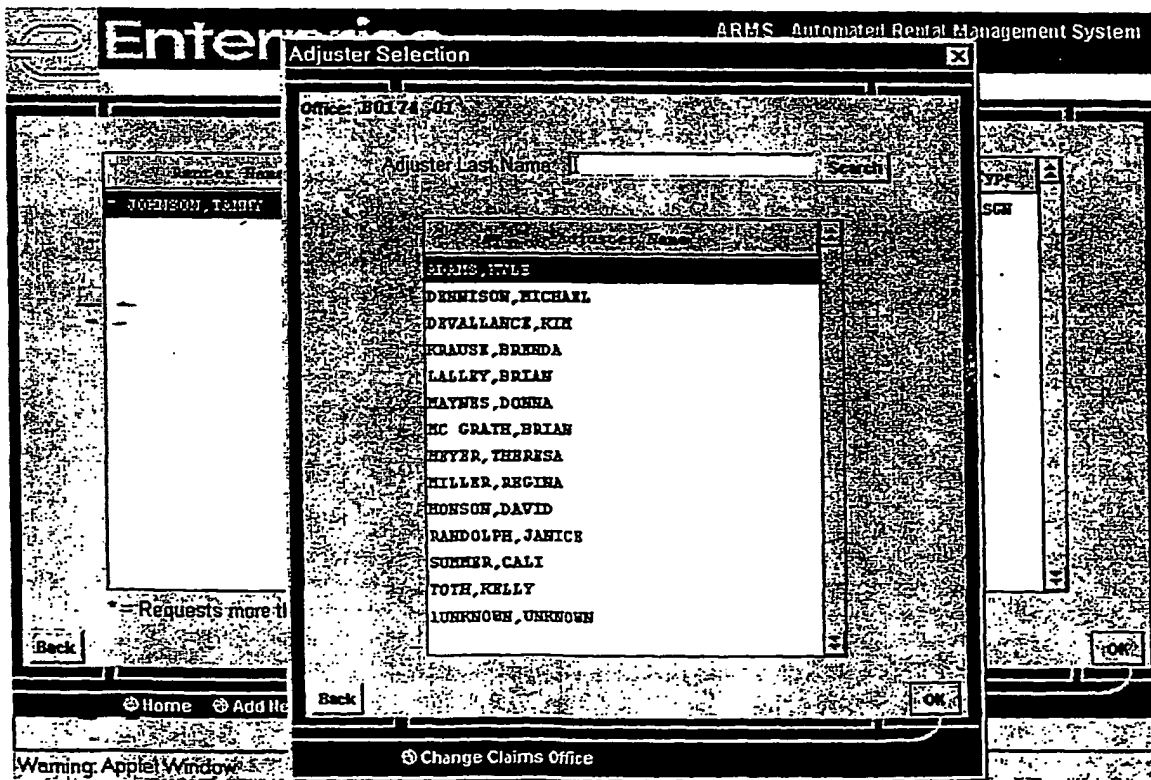
OR

Click to highlight the desired adjuster in the Adjuster name field, and click OK.

- 2) The Create Reservation screen appears (page 4).



Click Change Claims Office to find a new claims office.



## Create a Reservation - Rate Selection

This pop-up window appears when you click Rates from the Create Reservation screen (page 4).




The window shown below displays an *example* of vehicle classes and the associated rates. The rates displayed in this window are determined by the telephone number keyed in the **Phone Where Rental Is Needed** field on the Create Reservation screen. ARMS Web uses this telephone number to locate the Enterprise office closest to where the renter needs the car. The vehicle classes and rates that display are valid only at that particular Enterprise office.

This window also displays the Enterprise office location and phone number that you may wish to give to the renter.



*If there are no Enterprise offices found, based on the Phone Where Rental Is Needed field, the reservation will be sent to Claims Connection in St. Louis.*

- 1) Click to highlight the car class you wish to select, and click OK. ARMS Web automatically places the selected rate in the appropriate field and returns you to the Create Reservation screen (page 4).
- 2) Click Back to return to the Create Reservation screen without selecting a car class.



Claim Number:

Quick Rez

Renter Last Name:

Work Phone:  000  000

Home Phone:  000  000

Phone Where Rental is Needed:  434  4512

Authorized Amount:

Number of Days:

Vehicle Rate (Claimant):

Direct Bill Percent:  100 %

Message to Enterprise:

[Home](#) [Detail](#) [Message](#) [Change](#)

Rental Vehicle Class / Rate Selection

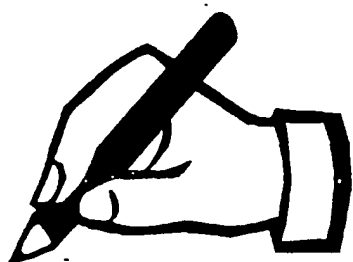
ENTERPRISE RENT A CAR - 0123

7730 BONHOMME AVENUE

CLAYTON, MO 63105

(314) 862-4885

CLASS	Description	DAILY RATE
EC	ECONOMY	18.99
CC	COMPACT	20.99
IC	INTERMEDIATE	22.99
SC	STANDARD	24.99
FC	FULL-SIZED	26.99
PC	PREMIUM	30.99
LC	LUXURY	36.99
SP	SPECIALTY	45.99
XP	PICKUP TRUCK	32.99
XV	VAN	35.99
XX	EXOTIC	65.99



# notes



# Change Reservation/Rental

Change Res/Rental



When should I use Change Reservation/Rental?



Use this section to extend a rental or change information on an existing reservation.

Authorization Status Definitions .....	1
Change a Reservation/Rental - Main Menu .....	2
Use a "Filter" Screen to Select a Reservation/Rental .....	3
Select a Reservation/Rental .....	4
Change an Unconfirmed Reservation/Ticket.....	5
Change an Unconfirmed Reservation/Ticket.....	6
<i>Renter, Vehicle, Repair Facility, and Claim Information</i>	
Change and Process an Unconfirmed Reservation/Ticket.....	7
<i>Authorization, Rental Delivery Information, and Message</i>	
Change and Process a Reservation or an Open Rental.....	8
Extend an Open Rental.....	9
Send a Message on an Open or Unconfirmed Reservation or Rental ...	10
Cancel a Reservation .....	11
Terminate an Open Rental Ticket .....	12

## Authorization Status Definitions

<b>RESERVATION</b>	A rental has been authorized, but the customer has not yet picked up a rental car or started a rental ticket/contract.
<b>OPEN RENTAL</b>	The customer has picked up a car and has opened a rental ticket/contract.
<b>DIRECT BILL REQUEST</b>	The customer has reserved or rented a car through Enterprise, and Enterprise is requesting that you confirm whether you will authorize the rental.
<b>EXTENSION</b>	An extension request is pending on an open contract.
<b>MESSAGE</b>	An incoming message is pending on an open reservation/rental.
<b>EXTN/MSG</b>	(Extension and Message) - An extension request, along with an additional message, is pending on an open contract.
<b>SENT</b>	You have sent an authorization to Enterprise, but the office has not yet sent back a confirmation.
<b>CLOSED</b>	(Closed Ticket) - The customer has returned the rental car and ended the contract.
<b>REJECT</b>	(Authorization Rejected) - An unconfirmed ticket or reservation has been denied authorization.
<b>CANCEL</b>	(Cancelled Reservation) - A previously authorized reservation has been cancelled.
<b>UNASSIGNED RESERVATION</b>	A direct bill request from Enterprise has yet to be assigned to an adjustor.

*The following status definitions apply only to those companies that use the ARMS Web Electronic Invoicing feature:*

**INVOICE** - The rental ticket has been closed and is awaiting approval for payment.

**REJECTED INVOICE** - Authorization for payment of the invoice has been denied.

**INVOICE RETURNED** - An invoice has been returned to the adjustor because the home office's system could not find a matching claim number and cannot issue payment.

**PAID** - The invoice has been approved and processed for payment.



The invoice is marked "PAID" in the ARMS Web system only. Your account will be credited upon receipt of payment.

## Use a "Filter" Screen to Select a Reservation/Rental

- 1) Key specific criteria to narrow the list of all reservations/rentals from which to choose.
  - \* For example, if you know the renter's claim number and date of loss, key that information in the appropriate fields.
- 2) Click Next.
- 3) If an exact match is *not* found, the Select Reservations/Rentals screen appears (page 4).  
If an exact match *is* found, the Change Reservation/Rental screen appears (page 5).

Any combination of information may be used to narrow the search for reservations/rentals.



Use the **Status** field in conjunction with the other information keyed to further narrow the search. If the **Status** field is left *blank*, the system searches *all* reservations/rentals.

Use the **Status** drop-down menu to search for open reservations only or open rentals only.



The **Last Authorized Day** field allows you to select rentals authorized up to and including the date provided.

You can also click any of the following option buttons (Appendix, page 1):

**Home** - Return to the main menu (page 2).

**Adjustor Lookup** - Display a list of adjustors (Appendix, page 5).

**Enterprise rent-a-car** Select Reservation/Rental Office: B0076 01

**Search Criteria**

Renter Last Name:  First:

\* Claim Number:

**Additional Search Criteria**

Adjuster Last Name:  ADAMS

Last Authorized Day:  / /

\* Date of Loss:  / /

Status:  All

Back Home Adjustor Lookup

## Change an Unconfirmed Reservation/Ticket

This Change Reservation/Rental screen displays when a match is found for an unconfirmed reservation/ticket. The status displays in the upper left-hand corner of the screen.

The information on each of the tabs (Authorization Information, Rental Location, Repair Information, Renter Information) depends upon the status of the reservation/rental.

- 1) Click on any of the tabs to alter renter, vehicle, repair facility, and/or claim information.
- 2) Click Finish to complete your change and return to the main menu.

You can also click any of the following option buttons (Appendix, page 1):

**Rates** - Display a list of vehicle rates.

**Home** - Return to the main menu (page 2).

**Detail** - Advance to the next Change Reservation/Rental screen (page 6).

**Message** - Send a message to an Enterprise office (page 10).

**Change Adjuster** - Change the adjustor handling the file (Appendix, page 5).

**Additional Charges** - View additional charges on the rental contract (Appendix, page 7).

**Cancel** - Cancel the reservation (page 11).

**Enterprise rent-a-car** Change Reservation/Rental  
Office: B0076 01 Adjuster: ADAMS, KYLE

Status: RESERVATION Renter: THOMPSON, MICHAEL Claim: 051287952

Date	Type	Description	Days	Rate	Total
10/25/99	P	AUTH CONFIRMED BY ENTERPRISE @ 14:13:07	0	\$	
10/25/99	R	RESERVATION NUMBER 152187	0	\$	
10/25/99	S	AUTHORIZATION SENT 14:13:04 5 DAYS @ 30.00/DAY	5	\$	150.00

Message to Enterprise: \_\_\_\_\_

**Authorization Information** | **Rental Location** | **Repair Information** | **Renter Information**

**Authorized Amounts**  
 Number of Days: 8  
 Vehicle Rate (Claimant): \_\_\_\_\_  
 Direct Bill Percent: 100 %

**Policy Coverage (Insured Only)**  
 Daily: 30.00  
 Maximum: 900.00

**Authorization Total: 150.00**

Back Finish

Rates Home Detail Message Change Adjuster Additional Charges Cancel

## Change and Process an Unconfirmed Reservation/Ticket

### Authorization, Rental Delivery Information, and Message

This Change Reservation/Rental screen displays when you click Next on the previous Change Reservation/Rental screen (page 6).

- 1) Key additional information, or key over any existing information you wish to change.



The Number of Days and the Vehicle Rate fields are *required* to confirm the authorization/reservation. If you do *not* key the number of days authorized, the reservation/ticket remains unconfirmed.

- 2) Click Finish to process the changes to this unconfirmed reservation/ticket and return to the Select Reservations/Rentals screen (page 4).

You can also click any of the following option buttons (Appendix, page 1):

**Rates** - Display a list of vehicle rates (page 5).

**Home** - Return to the main menu (page 2).

**Message** - Send a message to an Enterprise office (page 10).

**Change Adjuster** - Change the adjuster handling the file (Appendix, page 5).

**Additional Charges** - View additional charges on the rental contract (Appendix, page 7).

**Enterprise rent-a-car**

Change Reservation/Rental

Office: B0076 01 Adjuster: ADAMS, KYLE

Claim Number: 4512879624 Insurance: Insured

Authorized Information

Number of Days: 6 Vehicle Rate: 30.00 Direct Bill Percent: 100

Rental delivery information

Phone Where Rental Needed: 314 512 5000 Date Needed: / /

Message:

Back Finish

ⓧ Rates ⓧ Home ⓧ Message ⓧ Change Adjuster ⓧ Additional Charges



## Extend an Open Rental

This Change Reservation/Rental screen appears when a match is found for an open rental. The status displays in the upper left-hand corner of the screen.

To extend a rental,

- 1) Key the desired number in the **Number of Days to Extend** field.
- 2) Key a message if desired.
- 3) Click **Extend Rental**.
- 4) Click **Finish** to process the extension and return to the Select Reservations/Rentals screen (page 4). Click **Back** to return to the Select Reservations/Rentals screen (page 4) without processing any extension.



The next time you access this rental, the S/R message displays "EXTENSION SENT BY ADAMS,KYLE @ 10:46:21." The **Total** column will also reflect the extended number of days.

For an explanation of the buttons, please refer to page 8.

# Enterprise

## rent-a-car

Change Reservation/Rental

Office: B0076 01      Adjuster: ADAMS, KYLE

Status: OPEN RENTAL
Time: CLAIMANT
Claim: 100215832

Renter: ANTON
Vehicle: CHRIS

Date	S/R	Description	Days	Total
10/20/99	S	TERMINATION SENT BY ADAMS,KYLE ON 10/17/99	13	
10/15/99	R	TICKET 073082 OPENED 10/15/99 @ 08:00:00	0	10
10/14/99	R	AUTHORIZATION CONFIRMED BY ENTERPRISE @ 13:22:30	0	10
10/14/99	R	RESERVATION NUMBER 111395	0	10
10/14/99	S	AUTHORIZATION SENT 13:22:30 10 DAYS @ 20.99/DAY	10	10

Message to Enterprise:

Number of Days to Extend:  **Extend Rental**

**Authorization Information**

Authorized Amounts:

Number of Days	13
Vehicle Rate (Claimant)	20.99
Direct Bill Percent	100 %

**Rental Location**

**Policy Coverage (Insured Only)**

Daily

Maximum

**Repair Information**

**Repair Information**

Rental Start: 10/15/99

Authorization Terminated: 10/27/99

Authorization Total: 212.57

plus tax/surcbg

**Back**
**Finish**

• Rates
• Home
• Detail
• Message
• Change Adjuster
• Additional Charges
• Terminate

## Cancel a Reservation

The Cancel Reservation pop-up window appears when you click **Cancel** on the Change Reservation/Rental screen (page 5) if the status is "Unconfirmed Reservation" or "Reservation."

- 1) Key a **comment** that explains why this unconfirmed reservation or reservation is being cancelled.
- 2) Click **OK** to process the cancellation and return to the Select Reservations/Rentals screen (page 4). Click **Back** to return to the Select Reservations/Rentals screen (page 4) without cancelling the reservation.



The status changes from "Unconfirmed Reservation" or "Reservation" to "Cancelled."

# Enterprise

## rent-a-car

Change Reservation/Rental

Office: B0076 01     Adjuster: ADAMS, KYLE

**Cancel reservation**

Comment:

**INSURED HAS NO RENTAL COVERAGE**

Back    OK

	Days	Total
	7	7
09:42	0	6
	0	6
7	6	6

Back    Forward

**Authorization Amounts**

Number of Days: 7

Vehicle Rate (Claimant): Rates

Direct Bill Percent: 100 %

**Policy Coverage (Insured Only)**

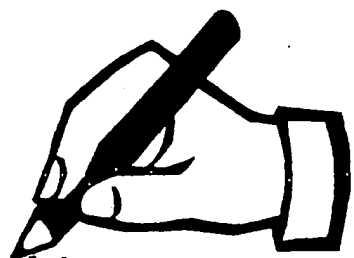
Daily: 30.00

Maximum: 900.00

Authorization Total: 210.00

Back
Forward

Rates
Home
Detail
Message
Change Adjuster
Additional Charges
Cancel

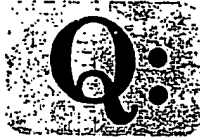


# notes



# Enterprise Requests-by Adjuster

## Requests-by Adjuster



When should I use Enterprise Requests-by Adjuster?



Use this section to respond to authorization requests for unconfirmed reservations/tickets and/or to extend rentals.

<b>Enterprise Requests-by Adjuster - Main Menu .....</b>	<b>1</b>
<b>Select a Reservation/Rental .....</b>	<b>2</b>
<b>Respond to Authorization Requests from Enterprise for an Unconfirmed Reservation (Direct Bill Request) .....</b>	<b>3</b>
<b>Review and/or Change an Unconfirmed Reservation (Direct Bill Request) .....</b>	<b>4</b>
<i>Renter, Vehicle, Repair Facility, and Claim Information</i>	
<b>Review and/or Change an Unconfirmed Reservation .....</b>	<b>5</b>
<i>Authorization, Rental Delivery Information, and Message</i>	
<b>Extend an Open Rental.....</b>	<b>6</b>
<b>Terminate an Open Rental Ticket .....</b>	<b>7</b>
<b>Send a Message on an Open or Unconfirmed Reservation or Rental .....</b>	<b>8</b>

## Enterprise Requests-by Adjuster - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjuster field.
- 3) Click Enterprise Requests - by Adjuster.
- 4) The Select Reservations/Rentals screen appears (page 2).



If you key an adjustor's last name in the **Handling for Adjuster** field, Enterprise requests for *only* that adjustor display on the Select Reservations/Rentals screen (page 2). Records appear in alphabetical order according to the renter name.

If you leave the **Handling for Adjuster** field *blank*, requests for *all* adjustors display on the Select Reservations/Rentals screen (page 2). Authorization requests appear in alphabetical order according to the **adjustor** name.

If you want to display requests for one particular adjustor, leave your name in the **Adjuster** field and input the other name in the **Handling for Adjuster** field.

Enterprise rent-a-car

Office: B0027 01 Adjuster: ADAMS

Handling for Adjuster: [Blank] (On All)

Renter's Last Name: [Blank]

Create Reservation

**Enterprise Requests by Adjuster**

Change Reservation/Rental

Enterprise Requests Unassigned

View Reservation/Rental

On Line Reporting

Process Approved Invoice

Setup and Maintenance

Logoff

## Respond to Authorization Requests from Enterprise for an Unconfirmed Reservation (Direct Bill Request)

This Change Reservation/Rental screen appears when you select an unconfirmed reservation/ticket on the Select Reservations/Rentals screen (page 2).

- 1) Key the number of days and the amount you are authorizing. This includes the daily rate, the maximum dollar amount allowed on the rental, and the number of days allowed.
- 2) Click Next to continue to the next Change Reservation/Rental screen (page 4). Click Back to return to the Select Reservations/Rentals screen (page 2).



If the claim type is Insured or Theft, the Number of Days field and the Policy Coverage fields must be filled in (required). If the claim type is Claimant, the Number of Days field, the Vehicle Rate field, and the Direct Bill Percent field must all be filled in (required).

You can also click any of the following option buttons (Appendix, page 1):

**Rates** - Display a list of vehicle rates.

**Home** - Return to the main menu (page 1).

**Message** - Send a message to an Enterprise office (page 8).

**Change Adjuster** - Change the adjuster handling the file (Appendix, page 5).

**Additional Charges** - View additional charges on the rental contract (Appendix, page 7).

**Cancel** - Delete any information updated on this screen and return to the previous screen.

**Enterprise rent-a-car** Change Reservation/Rental  
Office: B0027 01 Adjuster: ADAMS, RYLE

Status: UNAUTHORIZED OPEN RENTAL Type: CLAIMANT Claim#: 5897532181  
Renter: COOPER, ADAMS

NO.	DATE	TIME	DESCRIPTION	BY	STATUS
10/29/99	P	10:21:44	AUTHORIZATION REQUEST RECEIVED @ 10:21:44		0 0
10/29/99	R		Adjuster: Ryle Adams**		0 0

Message to Enterprise: \_\_\_\_\_

**Authorization Information** | Rental Location | Repair Information | Renter Information

**Authorized Amounts**

Number of Days	1
Vehicle Rate (Claimant)	22.99 Rates
Direct Bill Percent	100 %

**Policy Coverage (Insured Only)**

Daily	
Maximum	

**Authorization Total** 200  
plus tax/surchg.

Back

Rates Home Message Change Adjuster Additional Charges Cancel

## Review and/or Change an Unconfirmed Reservation

### Authorization, Rental Delivery Information, and Message

This Change Reservation/Rental screen appears when you click **Next** on the previous Change Reservation/Rental screen (page 4).

- 1) Review the screen to ensure that all information is accurate. If you need to make changes, key over the existing information.
- 2) Click **Finish** to send this authorization request to Enterprise and return to the Select Reservations/Rentals screen (page 2).  
Click **Back** to return to the previous Change Reservation/Rental screen (page 4).

You can also click any of the following option buttons (Appendix, page 1):

**Rates** - Display a list of vehicle rates.

**Home** - Return to the main menu (page 1).

**Message** - Send a message to an Enterprise office (page 8).

**Change Adjuster** - Change the adjuster handling the file (Appendix, page 5).

**Additional Charges** - View additional charges on the rental contract (Appendix, page 7).

**Enterprise rent-a-car** Change Reservation/Rental  
Office: B0027 01 Adjuster: ADAMS, RYLE

Claim Number: 887582141 Type: Claimant

Authorization information:

Authorized Adjuster:	Vehicle Class:	Policy Coverage (Insured Only):
Number and Name:	PC	Maximum:
Vehicle Rate (Claimant): 22.99 RATE	(Secondary Being)	
Deductible Percent: 100 %		

--- NOT SPECIFIED ---

PC - ECONOMY PC - STANDARD PC - FULL-SIZED PC - PREMIUM PC - LUXURY PC - SPECIALTY PC - EXOTIC

Message:

Rental delivery information:

Phone Where Rental Needed: 000 000 0000 Ext: 0000

Need Insured: No

Date Needed: 10/29/1999

Back Finish

Rates Home Message Change Adjuster Additional Charges

## Terminate an Open Rental Ticket

This Terminate Rental pop-up window appears when you click **Terminate** on a Change Reservation/Rental screen (page 6) if the status is Open Rental.

- 1) Key the last date for which you authorize payment. This date cannot be prior to the current date if the rental is already extended through today. If the date keyed is greater than the current date, ARMS Web automatically extends the rental through that date.



*You will not receive any more extension requests from Enterprise on a rental after that rental is terminated. However, you can still send extensions on rentals that you terminate.*

- 2) Click the box to indicate whether the renter has been notified of the last date for which this payment is authorized (required). If the box is left blank, this indicates that the renter has not been notified.
- 3) Key a comment to explain why you are terminating the open rental ticket.
- 4) Click **OK** to process the termination.



The status still displays as "Open Rental"; however, a message displays in the Message field stating that the rental has been terminated and by whom. If necessary, you can extend a rental past a termination (see page 6).

**Enterprise** Change Reservation/Rental

**01** Adjuster: ADAMS, KYLE

**Terminate rental**

Termination Date: 11/08/1999

Renter has been notified: ☐

Comment: CAR WILL BE FINISHED

Back OK

**Extend Rental**

**Authorization Information**

Authorized Amounts:

Number of Days	7
Vehicle Rate (Glamant)	26.99 Rates
Direct Bill Percent	100 %

**Policy Coverage (Insured Only)**

Daily Maximum:

**Rental Information**

Rental Start: 10/26/99

Last Authorized Day: 11/01/99

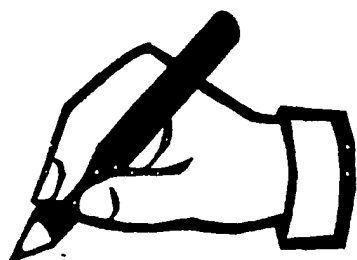
Authorization Total: \$68.93

plus tax/surchg

Back Finish

Rates Home Detail Message Change Adjuster Additional Charges Terminate



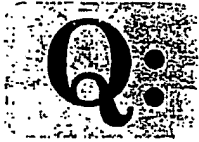


# notes

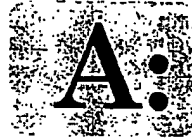


# **Enterprise Requests-Unassigned**

**Requests-Unassigned**



When should I  
use Enterprise  
Requests-  
Unassigned?



Use this  
section to  
assign  
Enterprise  
authorization  
requests to  
an adjuster  
or another  
claims office.

<b>Enterprise Requests-Unassigned - Main Menu .....</b>	<b>1</b>
<b>Select Unconfirmed Reservations/Rentals to Assign.....</b>	<b>2</b>
<b>View Authorization Request.....</b>	<b>3</b>
<b>Assign an Adjuster from Your Office .....</b>	<b>4</b>
<b>Assign an Adjuster from Another Office in Your Company .....</b>	<b>5</b>

## Enterprise Requests-Unassigned - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjustor field.
- 3) Click Enterprise Requests-Unassigned.
- 4) The Assign Reservations/Rentals screen appears (page 2).

**Enterprise**  
rent-a-car

ARMS Automated Rental Management System  
Home

Office: B0067 | 01 Adjustor: (Or ALL)

Handling for Adjustor: (Or ALL)

**Rental Activity**

- Enter's Last Name: [ ]
- Create Reservation
- Enterprise Requests by Adjustor
- Change Reservation/Rental
- Enterprise Requests Unassigned

**Rental Lookup**

- View Reservation/Rental
- On-Line Reporting
- Process Approved Invoices
- Setup and Maintenance

Logoff

## View Authorization Request

This Assign Reservation/Rental screen appears when you click and highlight a name and click OK on the previous Assign Reservations/Rentals screen (page 2).

- 1) Review the reservation/rental authorization request.
- 2) Click Assign to continue to the Adjustor Selection pop-up window (page 4), or click Back to return to the previous Assign Reservations/Rentals screen (page 2).




The Adjustor field is blank because this unconfirmed reservation has not yet been assigned to an adjustor.

You can also click any of the following option buttons (Appendix, page 1):

Home - Return to the main menu (page 1).

Message - Send a message to an Enterprise office (Appendix, page 4).



ARMS Automated Rental Management System  
 Assign Reservation/Rental  
 Office: B0067 01      Adjustor:

---

Status: UNASSIGNED/RESERVATION      Type: INSURED      Assign Number:

Renter: JOHNSON, BENNY  
 Address: 12345 ST. LOUIS, MO 63100  
 Work: 314-555-0000  
 Home: 314-555-0000

Rental: ENTERPRISE RENT-A-CAR  
 Address: 3850 LADUE ROAD  
 Airport: 314-555-0000  
 Location: PPGS Reservation/Assign Center

---

Rate Quoted on Rate of Rented Vehicle:

Repair Facility:

Process	S/P	Message	Date	Time
10/29/99	R	AUTHORIZATION REQUEST RECEIVED @ 10:48:04	0	0
10/29/99	R	Adjustor: Unknown*	0	0

Back
Assign

Home
Message

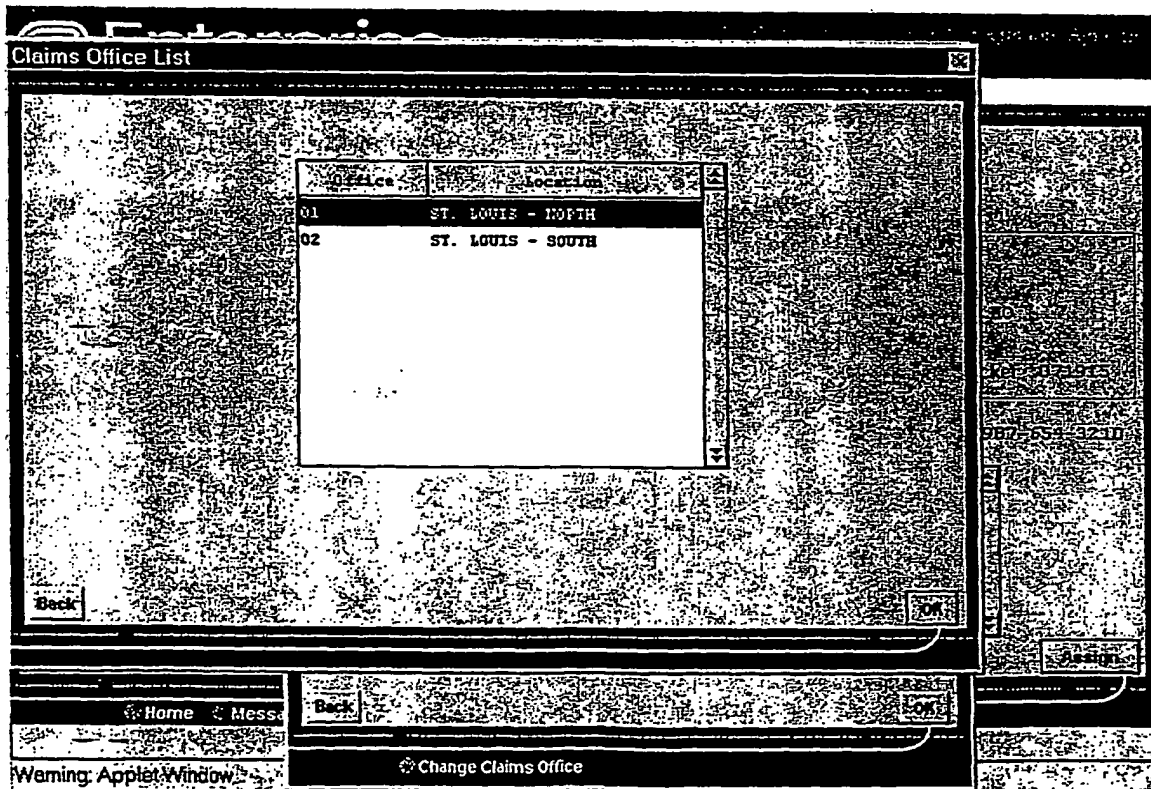
## Assign an Adjustor from Another Office in Your Company

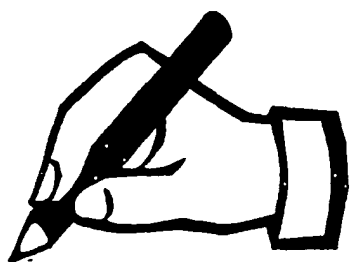
This Claims Office List pop-up window appears when you click **Change Claims Office** on the Adjustor Selection pop-up window (page 4).

- 1) Click to highlight the office you wish to select, and click **OK**.
- 2) The Adjustor Selection screen (page 4) from the office you chose appears.



Follow the instructions on page 4 to assign an adjustor from the office you selected.



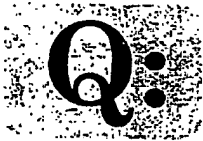


notes



# View Reservation/Rental

**View Reservation/Rental**



When should I  
use View  
Reservation/  
Rental?



Use this  
section to  
review a  
reservation  
without  
making any  
changes.

<b>View a Reservation/Rental - Main Menu.....</b>	<b>1</b>
<b>Use a "Filter" Screen to Select a Reservation/Rental .....</b>	<b>2</b>
<b>Select a Reservation/Rental .....</b>	<b>3</b>
<b>View a Reservation/Rental .....</b>	<b>4</b>
<b>View a Reservation/Rental .....</b>	<b>5</b>
<i>Renter, Vehicle, Repair Facility, and Claim Information</i>	
<b>View a Reservation/Rental .....</b>	<b>6</b>
<i>Authorization, Rental Delivery Information, and Message</i>	

## View a Reservation/Rental - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjuster field.
- 3) Click View Reservation/Rental.
- 4) The Select Reservation/Rental "filter" screen appears (page 2).

The screenshot shows the Enterprise rent-a-car main menu. At the top, the Enterprise logo is on the left, and the text "Home" is on the right. Below the logo, the text "rent-a-car" is displayed. To the right of "rent-a-car", the text "Office: B0117 01" is shown. Further right, the text "Adjuster:" is followed by a text input field containing "JONES". Below the "Adjuster:" field, the text "Handling by Adjuster:" is followed by a text input field containing "JONES". The main menu is divided into two columns of buttons. The left column contains buttons for "Create Reservation", "Enterprise Requests by Adjuster", "Change Reservation/Rental", and "Enterprise Requests Unassigned". The right column contains buttons for "View Reservation/Rental", "Online Reporting", "Process Approved Invoices", and "Setup and Maintenance". At the bottom left, there is a "Logout" button.



## Select a Reservation/Rental

This Select Reservations/Rentals screen appears under the following circumstances:

- ♦ More than one match is found for the information keyed on the Select Reservation/Rental "filter" screen (page 2).
  - ♦ No match is found for the information keyed on the Select Reservation/Rental "filter" screen (page 2).
- 1) Click to highlight a name in the Renter Name column, and click OK to select the renter whose reservation or rental you want to change.
  - 2) The first View Reservation/Rental screen appears (page 4).

You can also click any of the following option buttons (Appendix, page 1):

Home - Return to the main menu (page 1).

Refresh - Remove all updated information from the current page.

Renter Name	Phone Number	Location	Date	Status
REPAIRS, KELLY	1451558641	SUNSHINE, CALI	10/15/99	EXTN
ANTON, CHRIS	1459246832	ADAMS, KYLE	0/00/00	OPEN
BROWN, LUCILLE	3589249521	ADAMS, KYLE	0/00/00	OPEN
CLARK, MICHAEL	4519298216	ADAMS, KYLE	10/18/99	OPEN
CONNOLLEY, PAUL	1452658955	SUNSHINE, CALI	10/27/99	RESERV
CONRAD, KAREN	5897532141	ADAMS, KYLE	0/00/00	DB REQ
DAVIS, JOAN	5625874512	SUNSHINE, CALI	10/01/99	INV
EVANS, JOE	6578324681	ADAMS, KYLE	0/00/00	OPEN
GERLOFF, DANIELLE	4587145263	SUNSHINE, CALI	0/00/00	RESERV
GREEN, SARAH	25698745	SUNSHINE, CALI	0/00/00	DB REQ
HUNTER, RYAN	4512698754	SUNSHINE, CALI	10/01/99	CLOSED
JOHNSON, BETTY	1459287311	ADAMS, KYLE	0/00/00	OPEN
JOHNSON, TAMMY			0/00/00	UNASGN
JONES, MARK	1458796541	SUNSHINE, CALI	10/25/99	EXTN
JONES, PAULLEN	4516824672	ADAMS, KYLE	0/00/00	OPEN

\* = Requests more than 24 hours old

Back Home Refresh

## View a Reservation/Rental

### *Renter, Vehicle, Repair Facility, and Claim Information*


This View Reservation/Rental screen appears when you click Next on the previous View Reservation/Rental screen (page 4).

Click Next to continue to the next View Reservation/Rental screen (page 6) to view authorization, rental delivery information, and comments.

You can also click any of the following option buttons (Appendix, page 1):

**Home** - Return to the main menu (page 1).

**Additional Charges** - View additional charges on the contract (Appendix, page 7).



View Reservation/Rental  
Office: B0122 01     Adjuster: SUMMER, CALI

<b>Renter Information</b>			
Last Name: ABRAMS	First Name: KELLY		
Address: 123 MAPLE AVE.			
City/State: ST LOUIS	Zip: 63123		
Work Phone: 314 555 1212	Ext: 0000		
Home Phone: 314 555 8777			
<b>Vehicle Information</b>			
Make/Model: 95 CIVIC	Loss Vehicle: <input type="checkbox"/>		
Repair Facility: DAVE'S BODY SHOP			
Phone: 987 554 3210			
<b>Claim Information</b>			
Insured Last Name: [blank]			
Policy Number: [blank]			
Date of Loss: 10/25/1999			

Back
Next

Home
Additional Charges

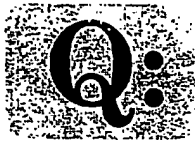


# notes



# On-Line Reporting

## On-Line Reporting



When should I use On-Line Reporting?



Use this section to review reports on rentals, broken down by category.

On-Line Reporting - Main Menu .....	1
Select Report View Options .....	2
Open Detail - By Adjustor .....	3
Open Detail - By Body Shop .....	5
Open Detail - By Office .....	6

EXHIBIT A



**E**nterprise **C**omputer **A**ssisted **R**ental **S**ystem

# Workbook



## TABLE OF CONTENTS

---

### *Introduction* *Page II*

---

A brief overview of the ECARS Workbook and Training System. Outlines functions that will be discussed within the workbook.

---

### *Getting Started* *Page III*

---

Sign On Procedures, Key Strokes, General Facts about RALPH, ECARS Training Menu.

---

### *Exercises*

---

Exercise 1	Reservations	Page 1-1
Exercise 2	Open a Ticket - Calendar Day	Page 2-1
Exercise 3	Open a Ticket - 24 Hour	Page 3-1
Exercise 4	Specials	Page 4-1
Exercise 5	Open Ticket Cross Reference	Page 5-1
Exercise 6	Correct a Ticket	Page 6-1
Exercise 7	Switching Units/Changing Rates	Page 7-1
Exercise 8	Callbacks	Page 8-1
Exercise 9	Closing a Ticket	Page 9-1
Exercise 10	Cash Management	Page 10-1

---

### *Error Messages* *Page VIII*

---

The most commonly seen error messages will be discussed along with corrective actions.

# **ECARS WORKBOOK**

## **INTRODUCTION**

---

The ECARS Workbook and Training System - AART01 - have been designed to teach the basic skills for working with ECARS - Enterprise Computer Assisted Rental System. Don't be alarmed if what you read in this workbook and what you see on the screen is slightly different. ECARS is constantly being improved and enhanced to work more efficiently for you.

### **Topics Covered**

---

#### **Reservations**

Opening rental tickets with various billing types

Ticket Cross Referencing

Correcting ticket information

Switching Units and Changing Rates

Callbacks

Closing rental tickets with various payment and billing types

Computerized CRS (Cash Receipt Summary) and deposits

### **Training**

---

Workbook requires approximately 3 hours to complete

Self Study

Work at your own pace

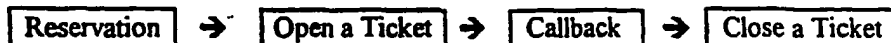
### **Workbook Format**

---

Each exercise guides you through an activity step by step

Most exercises build upon information covered in previous exercises

You will be able to see how different ECARS options are linked together:



Ranges from basic rental functions to refunds.

### **Numbers to Know**

---

The Rental Help Desk may be contacted at 1-800-416-8000. Specially trained ECARS personnel are available to answer questions and help with any problems you may encounter.

Network Services may be contacted at 1-800-416-8000. A team of Network Operators, who are RALPH experts, are available to correct any hardware problems experienced by the computer system.

***You are now ready to begin!***

### ***Rates/ Amounts/ Dates***

---

Use these examples for keying rates/amounts.

\$16.50	=	16.5	Field Exit
\$16.00	=	16	Field Exit
May 20, 1994	=	052094	

### ***Options and Choices***

---

When Yes or No answers are requested, key Y-Yes or N-No.

When several choices are given, key "X" or "I" in the selection field next to the item chosen.

### ***Names***

---

Names must be keyed in the correct format to help RALPH alphabetize names and distinguish between an individual or company name. Use these examples for keying names.

Individual:	Smith* Cindy*
	Smith* Cindy M*
Company:	Enterprise Rent-a-Car**
	Crawford Company**

(Key an asterisk \* by holding down the Shift key and pressing 8\*)

### ***Numeric Keypad***

---

Use the numeric keypad, located on the right side of the keyboard, to key mileage, rates, etc. The numeric keypad is designed in the same manner as a calculator; therefore, you may find it easier to work with.

### ***Important Keys***

---

There are several important keys on the keyboard that are essential to using ECARS.

**Reset**

When data is keyed incorrectly, that field will be highlighted and an error message will inform you of the problem. Press Reset and re-key the data.

**Enter**

Press Enter to send the information to be processed.

**F1,2,3...or  
Cmd 1,2,3..**

Command and/or Function keys are used to enter and exit various options. They are located on the top of the keyboard. Command/Function keys are displayed on the bottom of each screen.



**ECARS Training Menu**

To access the ECARS Training Menu, key AART01 at the Enter Request line **ENTER**. The following screen will appear.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMU-A
1 - Open A Ticket	11 - Reservations	<b>TRAINING SYSTEM</b>
2 - Correct A Ticket	12 - Callbacks	
3 - Switching Units/ Changing Rates	15 - Cash Management	
5 - Closing A Ticket		
(TICKET SERVICES)		
10 - Open Ticket X-Ref	Name _____	Or Ticket# 000000
Option# _____	D# 000000	Branch GM
Name _____	SSN# _____	Res# _____
DL# _____	ST/PROV _____	
Cmd1=Exit Cmd8=Lease Cust		

The ECARS Training Menu consists of the most used options in ECARS. Each of these options will have a corresponding exercise in this workbook.

- |  |  |
|--|--|
| <b>Option 1 - Open a Ticket:</b>                       | This option is used to open the majority of rental tickets in the office.  |
| <b>Option 2 - Correct a Ticket:</b>                    | This option allows you to correct and/or add information such as an additional driver, claim number, etc.  |
| <b>Option 3 - Switching Units/<br/>Changing Rates:</b> | This option is used to switch units and/or change rates.   |
| <b>Option 5 - Closing a Ticket:</b>                    | This option is used to close all rental tickets.   |
| <b>Option 10 - Open Ticket X-Ref:</b>                  | This option allows you to "look up" an open ticket to retrieve basic information.  |
| <b>Option 11 - Reservations</b>                        | This option is used to create, view, update, transfer, or cancel Branch Rental Reservations.   |
| <b>Option 12 - Callbacks</b>                           | This option is used to authorize or extend rental tickets by the type of callback. (For example Body Shop, Service Department, Insurance Company Adjustor, or Customer.) |
| <b>Option 15 - Cash Management:</b>                    | This option allows you to balance the cash summary and make deposits.  |
- 
- |              |   |
|--------------|---|
| <b>CMD 1</b> | Exit to Enter Request Prompt.                             |
| <b>CMD 8</b> | Lease Cust. Information about renting to Lease Customers. |

## **EXERCISE 1      RESERVATIONS**

Exercise 1 will guide you through Option 11 - Reservations - to create, view, update, transfer, or cancel Branch Rental Reservations. This option is very helpful in providing the branch(s) with better customer service, pickups, and deliveries.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMD-A
1 - Open A Ticket	(11) - Reservations	TRAINING SYSTEM
2 - Correct A Ticket		
3 - Switching Units/ Changing Rates	12 - Callbacks	
5 - Closing A Ticket	15 - Cash Management	
(TICKET SERVICES)		
10 - Open Ticket X-Ref    Name _____ Gr. Ticket# 000000 ----- Option# _____ D# 000000 Branch AC ----- Name _____ Home Phone # (____) _____ Res# _____ DL# _____ ST/PROV _____		
Cmd1=Exit Cmd8=Lease Cust		

On the ECARS Training Menu, key Option #11 **ENTER**. The Reservation Menu Screen will appear. (See example screen on the following page).

**NOTE:** You can key as little or as much information that is available when taking the reservation. Any information keyed into the reservation will automatically forward to the open ticket, saving you time when the renter is in the office picking up vehicle or waiting for a "Pick Up."

To update a reservation, any information keyed may be changed by keying directly over that which is displayed. Information may also be added or deleted if necessary.

## Category #1-Create a New Reservation

## Screen 1

1		BRANCH RESERVATION NUMBER 100003		Print (Y or N) <input checked="" type="radio"/> N		CCRS01-A	
Emp#		Pickup Branch FPGM		7			
2		Name (LAST*FIRST*)		SSN#			
3		PICKUP: Date Time		RETURN: Date			
W/in		P/up		Deliver		CWC	
4		PHONE: Home ( ) -		Office ( ) -		ext	
		Other ( ) -		Description			
5		T/B/D/R/C/O		Source Cust#		ID	
BILL TO:		Direct Bill (Y or N)		Credit Card		Cash/Check	
Bill To Cust#		or Name		(COMPANY**)			
Attention		Auth Until		Max Amt			
CLAIM#/Pol/Po		Date of Loss					
(C/Claimant, I/Insured, T/Theft)		Insured					
6		CAR TYPE: Class Type		F16		Comment	
RATE QUOTED: \$				Comment			
F1=Exit		F2=Cust List		F4=ID List		F5=Rates/Rules	
F9=Transfer		F12=Previous		F13=Insurance		F15=Res Notes	
F21=ID Rntr		F22=Clear Rntr ID				F23=More Keys	

The Reservation Screen is very similar to that of Option 1 - Opening a Ticket.

1. Key your five-character employee number at top of the screen.
2. Key Renter's Name (pretend you're the renter)-Last Name\*First Name\* and Social Security Number.
3. Key rental pickup, date/time, and rental return date. Use today's date for both pickup and return date. Key X to select either: W/in = Walk-in, P/up = Pickup, Deliver = Delivery, or CWC = Customer will call. **TAB RIGHT** to space provided for additional comments, if needed.
4. Key Renter's phone information. **TAB RIGHT** to description field for any comments, as needed.
5. Key Rental Type "I" - Insurance. The other rental types are B=Body Shop, D=Dealership, R=Regular, C=Corporate, or O=Other. Key "STATE" in the Source Cust# field. Press F2=Cust List. A Branch List appears. If there are no choices available, press F8=Group for a group list. Key "X" next to a State Farm Office. This automatically forwards to the Source ID List Screen (see page 2-3). Key "X" next to the first adjustor name.  
Key Y=Yes to set up a Direct Bill. Key "X" to select payment type: Credit Card, Cash/Check. For the Bill to Cust# use the F2=Cust List Window key, again. "X" your name on the list. Key "X" next to 999 UNKNOWN\*\* on the Source ID Screen. **NOTE:** If your name is not on the list, key 999999 as the Bill to Cust# and type your Last Name\* First Name\* on the name line. Key 999 as the Attention ID and key your name on the Attention Line. Key the Auth Until Date (use today's date), the Maximum Dollar Amount per Day, Claim/Pol/Po#, Date of Loss, Type of Loss, and Insured's Name.

Screen description continued on the following page.

### FUNCTION KEYS FOR OPTION 11 - RESERVATIONS

Listed on the bottom portion of each screen are several Function (F) keys. Each one performs a function that may be used while creating or viewing a Branch Reservation.

**F1 = Exit**

Allows you to cancel a new reservation. While viewing a reservation you can exit to the Reservation Menu Screen.

**F2 = Cust List**

Displays a list of customer names and numbers that are to be used for sourcing. See example window below.

MIDWEST		Branch List		SMS002A/C1	
GPRR 1520					
Position to description: _____					
Type options, press Enter.					
1=Select 5=Display					
Opt	Cust#	Description	State	Phone	
	G08433	ADAMS COUNTY CREDIT UNION**	IL	618-555-8877	
	G08246	AEROSPACE CREDIT UNION**	MO	314-555-0050	
	K00005	ALVIN JACKSON USED CARS**	MO	816-555-1277	
	K00171	AMERICAN AUTO MART**	IL	618-555-6688	
+-----+					
F2=Details F3=Exit F6=Branch F7=AAI F8=Group F10=State F11=All					
F12=Previous Roll=Forward/Back					

You can press F2=Details again to view a detailed branch list. The detailed list includes the customer name, address, and phone number. This is handy if you are trying to locate a specific customer who has multiple locations or if you need to contact the account in question.

Key "1" in the Opt (option) field to select a customer number. The screen automatically forwards to the contact screen. See example window on the next page.

**F8 = More Info**

Allows you to key additional renter information for the reservation.  
See example screen below.

<b>1</b>		MORE INFORMATION		CCRS05-A	
Renter Name KING* CATHY*		RES# 100963			
Renter Drivers License _____		State _____		Exp Date _____	
DOB _____		Height _____		Weight _____	
Employer _____		Eyes _____		Hair _____	
<b>2</b>					
Additional Driver (Y or N) _____		Driver Name _____			
Address _____		Age _____			
Drivers License _____		State _____		Exp Date _____	
<b>3</b>					
Out Of State (Y or N) _____					
<b>4</b>					
Bill-To Name _____		(Company Name**)			
Attention _____					
Address _____		Phone ( _____ ) _____		Ext _____	
City _____		State _____		Zip _____	
<b>5</b>					
Start Chgs. If Diff _____		Date _____		Time _____	
		Pickup Date _____		6/25/95 1:00 PM	
<b>6</b>					
Calendar _____		24-Hour _____		Specials _____	
<b>7</b>					
Rates: _____		.00 /Hour _____		.00 /Day _____	
Mileage _____		.00 /Mile After _____		0 /Day _____	
Drop. Chg. _____		Drop Location _____		.00 /Week _____	
Emp Waiver _____		.00 /Day _____		.00 /Month _____	
		Disc 0.0 _____		No Chg _____	
F5=Rates/Rules		F7=AAI		F12=Previous Enter=Update	

Remember, all information keyed here will automatically transfer to the Open Rental Ticket to save you and the renter time.

1. Key Driver's License Information and Current Employer's Name.
2. Key Additional Driver Information: Name, Address, Age, Driver's License #, State, and Expiration Date.
3. Key Y=Yes, or N=No, if the rental will be going out of state. If yes, key all associated states to which renter plans to travel.
4. Key "Bill To" Information: Company Name, Contact Person, Address, Phone Number, City, State, and Zip Code will be pre-loaded, if valid Customer # is keyed on the first Reservation Screen.
5. Pickup Date and Arrival Time will display, if keyed on the first Reservation Screen. Key new Date and Time to start charges, if applicable.
6. Key X to select billing type: Calendar or 24hr, and Specials, if applicable.
7. Key Rate Information and any Discounts, if applicable.

**NOTE:** This screen is edited in blocks. For example, if the Driver's License Number is entered then the State, Expiration Date, and Birth Date are required. Follow displayed ERROR MESSAGES to guide you as you go!

**F9 = Transfer**

Allows you to transfer a reservation to another Branch within your Group. This is helpful for location purposes and availability of units. It also allows you to transfer ARMS Reservations to another Group.

**F12 = Previous**

Return to previous screen. **NOTE:** If F12 is used, any updates just keyed will NOT be saved.

F14 = Credit Chk

**NOTE:** This function key is available and displays when F23=More Keys is pressed first. Allows access to enter valuable credit information when the reservation is created. This is helpful to determine money deposits or rental approval. See example screen below.

1		CREDIT CHECK INFORMATION		CCRS07-A RES# 100003	
Name	SMITH*LYNN*	(LAST*FIRST*)		5 Print (Y or N) <input type="radio"/> Y <input checked="" type="radio"/> N	
Street					
City	St	Zip	6 Credit checked? <input type="radio"/> Y <input checked="" type="radio"/> N		
Home#	( 000 ) 000 - 0000	Office#	( 000 ) 000 - 0000	ext 0000	
Other#	( 000 ) 000 - 0000				
SSN#	000 00 0000	DOB	Age 27		
Length of time at present address yrs mos					
2					
Curr Employer	Position		Length of Time		
Prev Employer	Position		yrs mos		
3					
Prev Street	City		St	Zip	yrs mos
Prev Street	City		St	Zip	yrs mos
4					
Spouse's Name	Employer		Position		yrs mos
Cmd7=AAI Cmd12=Previous Enter=Update					

Information keyed here will not appear on the Open Rental Ticket. This is reference information to be viewed within the reservation system.

1. Key Renter ID information: Name (Last Name\*First Name\*), Current Address, Phone Number(s), Social Security Number, Date of Birth, Age, and Length of Time lived at the above address (Years/Months).
2. Key Current Employer Name, Current Position Held, and Length of Employment (Years/Months). Repeat the process for previous employment, as needed.
3. Key Previous Address(s), and Length of Time lived at each location (Years/Months), if applicable.
4. Key Renter's Spouse Name, Current Employer Name, Current Position Held, and Length of Employment (Years/Months), if applicable.
5. Key Y=Yes, over default N=No, to print a copy of the Credit Check Screen.
6. Key Y=Yes if credit has been checked.

**NOTE:** Follow the policies set by your Group when creating a reservation. You can enter as much or as little of the credit information appropriate to the rental situation. The entire Credit Check Screen is optional. Ask your Rental Branch Manager to review situations applicable for Credit Check information to be taken.

**F21=ID Rntr**

Allows you to to locate previous renter information without exiting the reservation. See example screen below.

CENTRAL	ID Renter	FQS040A/1
DL# _____	St/Prov _____	
Home Phone# (____) ____ - ____		
Name (Last) _____	(First) _____	
<p>Note: Drivers license number and St/Prov or Home Phone number are required. Name is optional.</p>		
F12=Previous		

Key the Renter's Driver's License Number and State or Province OR the Home Phone Number (required). The Renter's name may also be keyed (optional). If more than one match is found, the Renter ID Selection screen appears. See example screen below.

CENTRAL	Renter ID Selection	FQS010A/1
Searched by: Phone Number (____) ____ - ____	D/L# _____	St/Prov _____
Position to: Name (Last) _____	(First) _____	
Type option, press ENTER		
1=Select      5=View		
Opt Name	Street Address	St Zip
- SMITH*KEVIN*	123 HARRISON	MO 63101 5684
- SMITH*RALPH*	467 MOLOKAI ROAD	HI 97895
- TAVERS*JOSEPH*	4223 VERLAINE AVENUE	WI 39667 7421
- TIPTON*ROBERT*	508 VIENNETTA DRIVE	LA 70460 6531
F3=Exit F12=Previous		

Search the screen to see if the customer is on the list. If the customer's name does not appear on this screen, key the driver's name (Last/First) in the Position to field. **ENTER**. The name keyed appears at the top of the list. Key "1" in the Opt (option) field to select a customer. **ENTER**. The customer's information is protected and loaded into the appropriate fields.

OR

Key "5" to view customer information in greater detail. **ENTER**. The Renter ID Detail screen appears. See example screen on the next page.

**ENTER**

to accept your Reservation.

\* \* \* \* \*

**You have just created a Branch Reservation!** For additional practice, open another reservation using the following information:

- *Use your favorite celebrity's name for the renter's name*
- *Do NOT key a Social Security Number*
- *Use today's date as the pickup and return date*
- *Make it a Bill To of "N"*
- *Make it a Callback type of "Cust"*

Be sure to key in all necessary information to make a complete reservation, using the above data plus your own chosen information.

\* \* \* \* \*

**Turn to the next page to learn how to view a reservation  
by Customer Name/Group  
(Category #2)**




**Category #3-View Reservation for Date/Group**

To view a Reservation by date, key in the Month/Day/Year in the space provided on the Reservation Menu Screen, #3. Today's date and your Group/Branch will already be pre-loaded. To view a different date or Group/Branch, key desired information over existing. A list of reservations will appear in sequential order - starting with the earliest Reservation associated with the date you requested. See example screen below.

RESERVATIONS FOR GPBR 7799				CCRS03-B	
Tuesday JUNE 21, 1994				Print (Y or N) <b>(N)</b>	
Total Reservations:				1	
Next Time		2			
Customer Name		Time	Res#	Car Type	Rental Type
FRANKLIN* BEN*			100501		
JENKINS* JOHN*		500 PM	A98761	ICAR	NRES
JONES* KIM*			100966	ECAR	DEL
Please meet her in the lobby.					
F1=Exit F7=AAI F12=Previous Screen Roll=Forward/Back					

This screen displays a list of all reservations for a Group/Branch or Group for a specific date. You can view a specific reservation by keying X next to the applicable Customer Name.

1. Displays Customer Name, Time of Arrival, Reservation Number, Car Type requested, rental Status, Rental Type, and Comments, if applicable. To view a specific reservation, key X on the line next to the Customer's Name.  to move cursor down the list of names, as needed. Press **ENTER** to advance you to the selected reservation(s) screen.

**NOTE:** This display can also be selected on the Reservation Menu Screen, #3 by Group. When only a Group is selected the reservation comments are NOT shown as in the above example.

**Reminder!** Scroll forward/backward to view all Customer Reservations. To make a revision to a reservation, key X next to the applicable Customer Name, press **ENTER** and make revisions, as needed.

2. Next Time: Key Next Time (Hour:Minutes/AM/PM) of reservations to be viewed. This is optional, as needed. **ENTER** to advance to the new time listings as requested.

3. To print a copy of the reservations listed, key Y=Yes, over default N=No.

**NOTE:** The function keys are discussed in detail, starting on page 1-5.

## EXERCISE 2      OPEN A TICKET - CALENDAR DAY

This exercise will guide you through Option 1 - Open a Ticket for a Calendar Day. Be sure to follow the format exactly as stated in this exercise. This ticket will be used again later on in the workbook!

If necessary, key AART01 **ENTER** to access the ECARS Training Menu.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMU-A
① - Open A Ticket	11 - Reservations	<b>TRAINING SYSTEM</b>
2 - Correct A Ticket		
3 - Switching Units/ Changing Rates	12 - Callbacks	
5 - Closing A Ticket	15 - Cash Management	
(TICKET SERVICES)		
10 - Open Ticket X-Ref    Name _____    Or Ticket# 000000		
1 - Option#	2 - D# 000000	3 - Branch AC
4 - Name _____	5 - Home Phone # (____) _____	7 - Res# _____
	6 DL# _____	ST/PROV _____
Cmd1=Exit    Cmd8=Lease Cust		

1. Option # Field:    Key Option Number 01.
2. D# 000000:    This field will remain blank; a ticket number has not yet been created.
3. Branch:    Your branch number will be displayed in the field. **←** to move the cursor to the next field.
4. Name    Key the first three letters of your last name in this field..
5. Home Phone #    Key your Home Phone Number in this field and/or (see number six).
6. DL#, ST/PROV:    Key your Driver's License number and State or Province.
7. Res#:    This field will remain blank for this exercise. A National Reservation Number or Branch Reservation Number is keyed in this field if one is known.

**ENTER** to accept the information keyed and advance to the next screen.

You should see a listing of all reservations for customers whose last name begins with the same three letters as were keyed. Locate and "X" the reservation you created for yourself. This information will forward to the ticket.

**CMD 1**    Exit to Enter Request Prompt.

**CMD 8**    Lease Customer. Information about renting to Lease Customers.

**NOTE:** The information that you had keyed into the reservation should have forwarded to the Open Rental Ticket. For those fields that you left blank, please fill them in appropriately.

### Section 2 - Renter Information

1	RENTAL TYPE I (I/B/D/R/C/O)	Source Cust #	2	ID	
Name	3	(Last*First*)			
Street	4				
City	5	ST		Zip	
Home Phone	( )		6		
Office Phone	( )		Ext	Employer	
Other Phone	( 000 )	000	-	0000	7 Description
Local Addr	8				

1. Key I - Insurance for Rental Type.
2. Use State Farm as the Source Customer for this exercise. Key "State" in the Source Customer # Field. Press F2=Cust List. The following screen appears.

MIDWEST	Branch List	SMS002A/ci
GEBR 1520		
Position to description: STATE		
Type options, press Enter.		
1=Select 5=Display		
Opt	Cust#	Description
	STF433	STATE FARM**
	STF246	STATE FARM**
	STF005	STATE FARM**
	STF171	STATE FARM**
		State Phone
		IL 618-555-8877
		MO 314-555-0050
		MO 816-555-1277
		IL 618-555-6688
+		
F2=Details F3=Exit F6=Branch F7=AAI F8=Group F10=State F11=All		
F12=Previous Roll=Forward/Back		

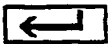
You can press F2=Details again to view a detailed branch list. The detailed list includes the customer name, address, and phone number. This is helpful when there are multiple locations for the customer. If there are no customers on the branch list, you can press F8=Group for a Group list, F10=State for a State list, or F11=All for a list including all customer numbers.

Key "1" in the Opt (option) field next to the first State Farm to select a customer number. The screen automatically forwards to the contacts screen. See example screen on the next page.

**Section 3 - Driver License**

<sup>1</sup> DL# 123456789	<sup>2</sup> ST MO Expires 062594	<sup>3</sup> DOB 032760	<sup>4</sup> SS# 555 55 5555
Height 6 02 Weight 175 Eyes BROWN Hair BLONDE			

1. Key the Driver License Number (no spaces are necessary).
2. **TAB RIGHT** and key the State abbreviation and Expiration Date.
3. Key Date of Birth.
4. If you keyed your SS# on the Training Menu Screen, it will be pulled forward. If not, key the SS#.
5. Key Height, Weight, Eye Color and Hair Color.



to move cursor to the next section, if necessary.

Screen 2 is broken down into three sections.

1	
2	Source Cust# 999999 ID 999 <b>COMPANY BILL TO</b> (Y/N) _ Cust# _ ID _ Attn _ Complete If Name _ (Company Name**) Cust # 999999 Street _ City _ ST _ Zip _ Phone ( _ ) _ - _ Ext _
3	<b>CLAIM/POL/PO#</b> _ Max Amount _ (C/Claimant, I/Insured, T/Theft) _ Insured _ Loss Date _ Car Sale Referral _ SHOP Cust# _ Name _ Attn _ Car Yr _ Make/Model _ Phone ( 000 ) 000 - 0000
F2=Cust List F3=Exit F4=ID List F5=Rates/Rules F7=AAI F8=Prewrite F10=CC Approval F11=CK Approval F12=Prev	

1. **Special Instructions:** If Special Instructions exist for the Source Customer Number, they will be displayed.

2. **Company Bill To:** This section requires a YES or NO answer. If yes, the Customer Number to be billed, the ID # and other company information are required.

3. **Claim/Pol/PO#:** This section requires the Claim, Policy or Purchase Order Number along with information regarding the Damaged/Stolen Vehicle, Insurance Rate and Shop Information.

**NOTE:** The function keys will be discussed in detail at the end of this exercise, page 2-14.

Screen 3 is broken down into four sections.

1					
2	Callback Type	B/S	Adj	Svc	Cust
3	Calendar Day	24 Hour Day	Specials		
4	Sales Tax %	Fuel Charge	Per		
	Drop Charge	Drop Location			
	GOVTSCHG	06%			
	AP/ACC	06%			
	LESERTAX	10%			
	ADDDLDRVR	2.00 per day	Additional Driver Indicated		
F3=Exit    F5=Rates/Rules    F8=Prewrite    F10=CO Approval    F11=Check Approval F12=Previous					

1. **Special Instructions:** If Special Instructions exist for your Bill To Customer Number, they will be displayed.

2. **Callback Type:** A "Callback" is a service offered to all customers. The rental branch personnel places calls to Body Shops and Service Departments to check the status of the customer's vehicle. Calls are then made to adjustors for extensions and to customers to pass along information (the car is ready, last day the insurance company will pay is XXX, etc.). This section requires the Callback Type to be marked with an "X". The Callback Type will help RALPH distinguish the callback list to which this ticket should be assigned.

3. **Calendar/24 Hour/ Specials:** You can decide here whether billing will be on a Calendar day or 24-hour basis. You can also go to the "Specials" screen by placing an X next to Special.

4. **Taxes and Additional Charges:** This section will display tax information, fuel charges, and additional charges that have been customized by Group/Branch. You can choose what charges apply by simply placing an X in front of the charge. You can also note if a drop charge applies.

**NOTE:** The function keys will be discussed in detail at the end of this exercise, page 2-14.

Screen 4 is broken down into six sections.

1 CAR TYPE REQUESTED _____ RATE QUOTED \$ _____.00 F5=Rates			
2 DATE OUT _____ Time _____ (Start Chgs If Diff Date _____ Time _____ Cust Ext Date _____ Time _____ Auth Until Date 000000			
3 ECAR # _____ License # _____ Or Serial # _____ Not on File _____			
RATE _____ .00 /Hour _____ .00 /Day _____ .00 /Week _____ .00 /Month Discs _____ MILEAGE _____ /Mile After: _____ /Day _____ /Week _____ /Month 4. _____ No Charge			
DW _____ .00 /Day \$136.34 Est Charges PAI _____ .00 /Day SLP _____ 5.99 /Day			
5 DEPOSITS Cash _____ .00 Check _____ .00 CC _____ .00			
6 CALLBACK NOTE _____ Update Code _____ Emp# _____ Emp# if Different _____ F3=Exit F5=Rates/Rules F6=Units Avl F7=AAI F8=Prewrite F9=Unit Pend F10=CC Approval F11=Ck Aprvl F12=Prev F16=Est Chgs			

1. Car Type Requested/Rate Quoted: This information will be pulled forward from the Branch or National Reservation. If a Car Type was not requested, or a rate was not quoted, NONE will be displayed in this field.

2. Date Out/Current and Customer Ext: Today's Date and Time will be supplied by Ralph. Current Ext Date appears if you placed an "X" in B/S, Adj or Service Callback. Customer Ext date appears if you placed an "X" in Customer Callback.

3. ECAR #: This section requires an ECAR Number and License or Serial Number to be keyed. You can key this from the key tag or by using F6=Units Available. If the unit information is not available, use F9=Unit Pending. The ticket will print, but you must complete this information, when available, using option 9 on the ECARS Main Menu, "Complete a Ticket".

4. Rate: This section requires information regarding the daily rate, along with the mileage limitations and charges. Estimated Charges, Damage Waiver, PAI and SLP daily rates are also included in this section. All rates are examples only. Please check with your Group for correct rates.

5. Deposits: This section requires information regarding the deposit to be taken at the time of the rental and how it was received.

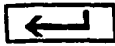
6. Callback Note: Key a brief note in the Callback Note field, as necessary. This note pulls forward to the Rental Ticket for reference.

**NOTE:** The function keys will be discussed at the end of this exercise, page 2-14.

**Section 5 - Deposits**

DEPOSITS	Cash	100	Check	.00	CC	.00
----------	------	-----	-------	-----	----	-----

Key 100 **FIELD EXIT** in the Cash field to designate a \$100 Cash deposit.



to move cursor to the next section.

**Section 6 - Callback Note**

CALLBACK NOTE	1
2 Update Code	Emp# 55555 Emp# if Different 00000

1. Key a brief note, as necessary.
2. Key your Update code and your 5-digit Employee Number.

**ENTER** to accept information keyed. Correct any errors, if necessary. The ECARS Training Menu will be displayed when ticket is accepted.

**Congratulations!**

You have just completed your first rental ticket.  
In the "live" ECARS program, the completed rental ticket would print from the DR Ticket Printer.



### EXERCISE 3      OPEN A TICKET - 24 HOUR

This exercise will guide you through Option 1 - Open a Ticket for a 24 hour clock. Be sure to follow the format exactly as stated in this workbook. This ticket will be used again later!

If the ECARS Training Menu is not still on the screen from the previous exercise, key AART01 on the Enter Request line **ENTER**. The ECARS Training Menu will appear.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMD-A
① - Open A Ticket	11 - Reservations	Training System
2 - Correct A Ticket	12 - Callbacks	
3 - Switching Units/ Changing Rates	15 - Cash Management	
5 - Closing A Ticket		
(TICKET SERVICES)		
10 - Open Ticket X-Ref	Name _____	Or Ticket# 000000
1 - Option#	DL# 000000	Branch GM
2 - Name	3 - Home Phone # ( )	Res#
	DL#	ST/PROV
Cmd1=Exit Cmd8=Lease Cust		

1. Option Field:      Key Option Number 01.
2. Name:              Key the first 3 letters of your favorite celebrity's last name in this field.
3. Home Phone #:      Do NOT key a Home Phone Number or Driver's License Number this time.

**ENTER** to accept the information keyed and advance to the next screen.

Locate and "X" your favorite celebrity's reservation you previously created. This information will forward to the ticket.

**CMD 1**      Exit to Enter Request Prompt.

**CMD 8**      Lease Customer. Information about renting to Lease Customers.

## Section 4 - Additional Driver

1		2	
ADDITIONAL DRIVER (Y/N)	Y	Driver Name	TIM HILLMAN
Age	25		
Addr	3	Lic#	4
St		Exp	

1. Key Y - Yes for an additional driver. Key the name of a family member or friend.
2. **TAB RIGHT** to move the cursor to the next field. Key the Age of the additional driver. If you do not know the exact age, but know they meet the age requirement, key 25
3. Key Additional Driver Address.
4. **TAB RIGHT** to move cursor to next field. Key Additional Driver License Information.  
**←** to move cursor to the next section, if necessary.

## Section 5 - Out of State

OUT OF STATE (Y/N)	Y	ILLINOIS
--------------------	---	----------

Key Y - Yes, and key the State(s) to which the renter plans to travel.

Press **ENTER** to accept the information keyed and advance to Screen 2. RALPH will scan the information you have keyed from top to bottom. If any errors exist, or if a field has been left blank, an ERROR MESSAGE will be displayed. This message will explain exactly what needs to be corrected and the cursor will be placed at that field. See the section on ERROR MESSAGES for more information.

Make any necessary corrections and press **ENTER** again. Repeat this procedure until Screen 2 appears.

## Section 2 - Callback Type

Callback Type	B/S	Adj	Svc	Cust
---------------	-----	-----	-----	------

Key an "X" to mark this ticket as a Customer Callback.

## Section 3 - Calendar / 24 Hour / Specials

Calendar Day	24 Hour Day	Specials
--------------	-------------	----------

**TAB RIGHT**

to move cursor to next field an Key an "X" in the selection field for 24 hour.

## Section 4 - Taxes and Additional Charges


Sales Tax %	Fuel Charge	Per
Drop Charge	Drop Location	
GOVTSCHG	06t	
AP/ACC	06t	
LESERTAX	10t	
ADDLDRVR	2.00 per day	Additional Driver Indicated
F3=Exit	F5=Rates/Rules	F8=Prewrite
F12=Previous	F10=CC Approval	F11=Check Approval

The Sales Tax/Surcharge and Fuel Charges will be preloaded by Ralph. The fields for Drop Charge and Drop Location will remain blank for this exercise. You may see a few other charges in this section. These charges are Group/Branch specific and have been customized by your Group Business Manager.

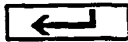
**ENTER** to advance to Screen 4. If any errors exist, correct them as you did on the previous screen.

## Section 4 - Rate

RATE 1	.00 /Hour	.00 /Day	.00 /Week	.00 /Month	Disc: 8
MILEAGE 2	/Mile After:	/Day	/Week	/Month	
	No Charge.				
3	DW	.00 /Day			
	PAI	.00 /Day			
	SLP	.00 /Day			

1. Key 6 **FIELD EXIT** in the Hourly Rate field. Key 24.95 **FIELD EXIT** in the Daily Rate field. The Weekly and Monthly Rate fields will remain blank for this exercise. The field for Discount will remain blank also. 

2. Key 25 in the Mileage Charge field. Key 100 in the Per Day Mileage Allowance field. Weekly, Monthly and No Charge will remain blank.



to move cursor to Damage Waiver/Day Field.

3. Key 9 **FIELD EXIT** in the Damage Waiver field. Key 1 **FIELD EXIT** in the PAI field.

**FIELD EXIT** over the SLP amount to remove it.

## EXERCISE 4      SPECIALS

Exercise 4 will guide you through opening a ticket with a Weekend/Holiday special rate. You will be the renter.

Use Option 1 - Open a Ticket and complete Screens 1,2 and 3. In the section "Calendar/ 24 Hour/ Specials,"

"X" 24 hour and Specials **[ENTER]** . The following screen will appear.

WEEKEND SPECIAL RATES					
1.	START SPECIAL	Date 051195	Time 0925 AM	OR	Date _____ Time _____
	END SPECIAL	Date _____	Time _____		
2.	DAILY SPECIAL	Rate _____ / Day			
		Miles _____ / Day	OR		No Charge
	***** OR *****				
	PACKAGE SPECIAL	Rate _____ / Pkg			
		Miles _____ / Day	OR		No Charge
F3=Main Menu F5=Rates/Rules F7=AAI F12=Previous					

Two Options are available for Special Rates.

Option 1 - Daily Special.

Option 2 - Package Special (one price charged for a specific number of days and miles).

1. Key an X in the slot to the left of the current date and time. If you wanted to start the special at a date/ time other than the current date/time, you could key the appropriate date/time in the field provided. Key the date and time that the special will end.

The Start and Stop Dates are very important, especially if the renter has the vehicle for any length of time before or after the Special Rates apply.

For this exercise, we will use a daily rate of \$19.95 per day/100 miles per day.

2. Key the Rate and Mileage information in the appropriate fields in the section for Daily Special.

**[ENTER]** Screen 4 for a 24 hour rental will appear.

F3=Main Menu.	Return to the ECARS Main Menu.
F5=Rates/Rules.	Allows access the Rates/Rules Screen.
F7=AAI.	Allows access to inquiry programs.
F12=Previous	Returns to the previous screen.

## EXERCISE 5      OPEN TICKET CROSS REFERENCE

Exercise 5 will show you how to look up an open Daily Rental Ticket by Name or Ticket #. This option is very helpful for identifying Gpbr, Unit Number and License Number information.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMU-A
1 - Open A Ticket	11 - Reservations	TRAINING SYSTEM
2 - Correct A Ticket		
3 - Switching Units/ Changing Rates	12 - Callbacks	
5 - Closing A Ticket	15 - Cash Management	
[TICKET SERVICES]		
10 - Open Ticket X-Ref	<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> <div>2.      A Name <u>SMITH</u></div> <div>B Or Ticket# <u>000000</u></div> </div> </div>	
<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> <div>Option# <u>1</u>      DR# <u>000000</u>      Branch <u>GM</u></div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div>Name <u>                    </u></div> <div>Home Phone # ( <u>      </u> ) <u>      </u></div> <div>Res# <u>          </u></div> </div> <div style="display: flex; justify-content: space-between; margin-top: 5px;"> <div>DL# <u>                    </u></div> <div>ST/PROV <u>          </u></div> </div> </div>		
Cmd1=Exit    Cmd8=Lease Cust		

1. Key Option # 10. **TABLE LEFT** to position the cursor in the selection field to the right of the Option Number. See the box marked on the example.
2. The customer's last name is keyed in Field A. If the ticket number is known, but not the customer name, key the ticket number in Field B. For this exercise, key your last name in Field A **ENTER**. The following screen will appear.

NEXT CUSTOMER		Key customer's name here to look up another ticket		CCRA14
Customer Name	GPR	Ticket	Unit #	License #
SMITH* ROBERT*	PPGM	999007	P00224	P00224
Cmd1=Return    Cmd3=Main Menu    Cmd7=AAI				

The Open Ticket Cross Reference displays the customer's Name, Gpbr, DR ticket number, Unit # and License #.

You may look up another ticket from this screen. Key your favorite celebrity's last name in the section marked Next Customer and press **ENTER**. RALPH will locate that name and display the information.

Write down the numbers of all the tickets you created. Return to the Training Menu by pressing

**CMD 1**

## ***EXERCISE 6      CORRECT A TICKET***

Option 2 - Correct a Ticket, allows you to correct/add/delete information on an open Daily Rental Ticket. This option is most commonly used to add claim information, shop information, additional drivers, etc. This option is NOT used to switch units or change rates.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CORTMU-A
1 - Open A Ticket	11 - Reservations	TRAINING SYSTEM
② - Correct A Ticket	12 - Callbacks	
3 - Switching Units/ Changing Rates	15 - Cash Management	
5 - Closing A Ticket		
(TICKET SERVICES)		
10 - Open Ticket X-Ref	Name SMITH	Or Ticket# 000000
Option#	DL# 999007	Branch GM
Name	Home Phone # ( )	Res#
DL#		ST/PROV
Cmd1=Exit    Cmd8=Lease Cust		

1. Key Option # 02.

2. Key the Daily Rental Ticket Number (use the lowest ticket number from Exercise 5 - Open Ticket Cross Reference).

**ENTER** to accept the information and advance to the Correct a Ticket screen.

## Screen 2

Current Rate Effective On 06/13/97

Bill-To

Attn

Claim/POL/PO#

Loss Date 000000

(C/Claimant, I/Insured, T/Theft)

Insured

SHOP:

Cust#

Name

Attn

Car Year 93

Make/Model BUICK/REATTACNV

Phone# ( 000 ) 000 - 0000

Surcharge \$ 5.850

Fuel Charge

5.00 Per RENTAL

Drop Charge 5.00

Drop Location

Airport Access 1.000 \$

SLP

2.00 /DAY

ADD DRIV

1.00 /DAY

Emp#

F2=Cust List F3=Exit F7=AAI F12=Previous

Screen 2 includes the sections:

•Current Rate Effective Date

•Claim/Policy/PO#

•Additional Charges

•Employee Number

Any information may be changed by keying directly over what is displayed. Information may also be added or deleted if necessary.

Add a \$5.00 drop charge and key the Gpbr Number of another branch in your Group as the drop location.

Key your 5-digit Employee Number.

**ENTER** to accept information.

**F2** Customer List.

**F3** Exit to Main Menu.

**F7** AAI. Access Inquiry programs.

**F12** Previous Screen.



## **EXERCISE 7      SWITCHING UNITS/CHANGING RATES**

This exercise will show you how to switch units and/or change rates on an open Daily Rental Contract. Use the ticket # from Exercise 3 - Open a Ticket, 24 Hour.

On the ECARS Training Menu Screen:

1. Key Option # 3.
2. Key the DR Ticket Number.

**ENTER** The following screen will appear.

1	D# 999013	Renter SMITH* ROBERT*	RA03A
<b>SWITCHING UNITS</b>			
UNIT 1	DC0232	DATE/TIME OUT 022296 0835 AM	START MILES 45000
		DATE/TIME IN 022296 0530 PM	END MILES 45101
Old unit returned here ___ or to ___			
New unit	License #	OR Serial #	Not on file
New unit date out	Time out		
2			
<b>CHANGING RATE-1</b> Original: 2/22/94 8:35 AM Current: 2/24/96 1:31 PM			
New rate effective date			
Calendar day	24 Hour Day X	Special	Discount 00%
Rate	6.00 /Hour	24.95 /Day	.00 /Week .00 /Month
Mileage	8.25 /Mile after 100 /Day	0 /Week	0 /Month X No Charge
Damage waiver/Day	9.00	PAI/Day	1.00
Cmd1=Return Cmd4=Units Avail Cmd5=Spec Cmd6=Rate Cmd7=AAI Cmd10=Prev Miles			

The screen is divided into 2 major sections.

1. **Switching Units:** The portion of the screen above the current date and time is used to switch units.
2. **Changing Rates:** The lower portion of the screen, below current date and time, is used to change rates.

Either one or both options may be used.

**NOTE:** The command (function) keys will be discussed at the end of this exercise, page 7-4.

**CHANGING RATES**

CHANGING RATE-1				Original: 6/01/94	8:35 AM	Current: 6/02/94	1:31 PM
1							
New rate effective date _____				Time _____			
Calendar day _ 24 Hour Day X				Special _____		Discount 00%	
2							
Rate		6.00 /hour	24.95/day	.00/week	.00/month		
Mileage @25 [ /mile after		100/day	.0/week	0/month	N/C		
3							
Damage Waiver /day		9.00	PAI/day	1.00			
Cmd1=Return Cmd4=Units Avail Cmd5=Spec Cmd6=Rate Cmd7=PAI Cmd10=Prev Miles							

1. Key the Date and Time the rate change becomes effective. Calendar Day, 24 hour and Special may be changed by keying X, if necessary. A Special Rate may be added/deleted, e.g.: daily rate or package rate. Do not update that information for this exercise.
2. Change the Daily Rate to \$32.99 per day, mileage will stay the same.
3. Damage Waiver and PAI may be added/deleted/changed.

**ENTER** to accept information.

***Congratulations!***

You have finished learning Exercise 7 - Switching Units/Changing Rates. Do NOT press **ENTER**. Turn to the next page and review the "Command" keys applicable to this section.

Cmd6=Rate

Displays current rental Rate History on top portion of screen, followed by previous rate history information: Date and Time charges started, Rate/per Day, Mileage (unlimited or charged), Damage Waiver or PAI purchased.

RATE HISTORY		Ticket #	PPGM	CCRH1R-D
C	Start Charges	02/20/94	10:00 AM	
U	Current rate	6.00/Hour	32.99 /Day	
R	Free Miles	100 /Day		
R	Damage Waiver	9:00 PAI 1.00	Excess Miles At 25 MI	¢
-----				
	Start Charges	02/22/94	08:30 AM	
	Rate	6.00/Hour	24.95 /Day	
	Free Miles	100/Day		
	Damage Waiver	9:00 PAI 1.00	Excess Miles At 25 MI	¢
-----				
Cmd1=Return Enter=Previous Screen				

Cmd7=AAI

Allows access to the Inquiry Programs.

Cmd10=Prev Miles

Allows you to change previously keyed mileage.

## EXERCISE 8      CALLBACKS

Exercise 8 will guide you through Option 12 - Callbacks - to view or update Open Rental Tickets and Reservations for your Branch or for another Branch on your machine. Also, the callback history will remain with the ticket after it has been closed. This is helpful if you have an adjustor or customer who has a question about a closed rental ticket.

ECARS - ENTERPRISE COMPUTER ASSISTED RENTAL SYSTEM		CCRTMU-A
1 - Open A Ticket	11 - Reservations	TRAINING SYSTEM
2 - Correct A Ticket	12 - Callbacks	
3 - Switching Units/ Changing Rates	15 - Cash Management	
5 - Closing A Ticket		
(TICKET SERVICES)		
10 - Open Ticket X-Ref	Name _____	Or Ticket# 000000
Option# _____	D# 000000	Branch AC _____
Name _____	Home Phone # ( _____ ) _____	Res# _____
DL# _____	ST/PROV _____	
Cmd1=Exit Cmd8=Lease Cust		

On the ECARS Training Menu, key #12 on the line following Option #. **ENTER**. The Callback Menu Screen will appear. (See example screen on following page).

**NOTE:** It is very important that you pay attention to detail while keying or updating all callback information. It will eliminate problems if you are careful at the start.

Turn to the next page and learn all about the Callback System.  
Good Luck!



Now, let's take a closer look at Section 1 - Review Tickets Needing Extensions. To start, key X to select Body Shop calls on the Reservation Menu Screen. **ENTER** to prompt the associated screen, shown below.

### Section 1 - Body Shop

BODY SHOP CALLS				5 Print (Y or N) <b>N</b>
1 NEXT SHOP	2 SHOP#	4	5 # OF CALLS	
AAA AUTO BODY	AA2134		007	
ABC BODY SHOP	999999	999-999-9999	003	
ALFRED'S AUTO BODY	123456	888-888-8888	003	
ALLEN'S AUTO BODY		777-777-7777	002	
BILL'S REPAIR PLACE		889-898-9898	001	
DODGE WORLD		666-666-6666	002	
ELCO CHEVROLET		333-333-3333	006	

Cmd1=Exit Cmd3=PAI Roll=Forward/Back

The Body Shop Screen lists all body shops alphabetically, including phone numbers and the number of calls to be made to that shop.


1. Next Shop (optional) - Key the Shop Name you chose when you opened the Calendar Day ticket. Press **ENTER**. The next list will appear. - OR -
2. Shop Name - Key X by the Shop Name you had entered.  to move cursor down the list of names, as needed. Press **ENTER**. The Body Shop Selection Screen will appear, listing customers who are having their cars repaired at that particular shop. See example screen on the following page.
3. Shop Number - Displays shop Customer Number.
4. Shop Phone Number.
5. Number of Calls - Displays number of calls to be made to Shop.
6.  to move cursor to top right of screen. To Print a copy of the Body Shop Callback Screen, key Y=Yes over default N=No.


**Reminder!** Use your roll-up and roll-down keys to scroll backward or forward through Shop listings.

## Section 1 - Body Shop Detail Screen

BODY SHOP CALLBACK DETAIL - D100096		CB50-1
DATE OUT 06/10/97 8:19AM	<sup>1</sup> CURR EXT 06/15/97	CUST EXT 06/15/97
CUSTOMER DAVIS* TOM*	HOME PHONE# 333-333-3333	
RATE 15.00 /DAY	OFFICE PHONE# 333-343-3434	EXT 2323
	OTHER PHONE# 333-343-3435	
BILL-TO Y BILL-TO CUST# 999999	<sup>2</sup> SHOP# 9999999	FRANK'S AUTO BODY
BILL-TO NAME LENKMAN* JANE*	PHONE# 314-456-9870	
PHONE# 212-212-2121 EXT7896	ATTN Bob Brown	
ID/ATTENTION 999 JANE LENKMAN	YEAR 94 MAKE/MODL ACURA INTEGRA	
CLAIM/POL/PO#	LOSS DATE 000000	
MAX AMOUNT 25.00	INSURED	
NEW INFO EMP#	<sup>3</sup>	
06/10/94 11:41 AM RENTAL EXTENDED UNTIL 06/13/94 BY TS 74024		
06/13/94 9:00 AM PROBLEM GETTING PAINT TO DRY 75026		
STATUS IN X DO OK	<sup>4</sup>	<sup>5</sup> CALLBACK TYPE X B/S X ADJ SVC CUST
EXTENSION DATE BY	<sup>6</sup>	
LAST DAY BY	<sup>7</sup>	
F2=Cust List F3=Exit F4=ID List F5=Open Ticket F6=Chgs To Date F7=AAI F8=Update Tkt F9=Callback Notes F10=Receipt Depos Roll=Fwd/Back F12=Previous F13=Insurance F14=Credit Check F15=Reservation		

Any background information you need to know about the renter is available on this detail screen.

1. Customer Information - Displays Ticket Number, Rental Date and Time Out, Current approved Extension Date, Renter's Name, Renter's home/office/other Phone, plus Rate the renter is being charged.
2. Bill-to Information - Displays the Bill-to Customer Number, Company Name, Phone Number and Extension, ID Number, Attention Name, Claim/Policy/P.O. Number, and Max Amount Bill-to Customer Number will pay. Also, Repair Shop's Customer Number and Name, Phone Number, Contact Person's Name at the Shop, Year/Make/Model of renter's vehicle, Loss Date, and Insured's Name is listed  to move cursor through the fields. To change ticket information, press F8=Update Tkt, then key new information over the existing information.
3. New Information - Displays all callback information to date. This is the *most important* and utilized part of the Callback System. RALPH automatically displays a record of all previous callback messages, date and time call was made, and employee number of the person who made the call. This record remains in the callback history of the closed ticket, for future reference. Use roll-up and roll-down keys to scroll backward and forward through the previous records, as needed.

**NOTE:** To enter *new* information received from a current call,  to move cursor, key your employee number in the "EMP#" field highlighted above. Key new information, "Body Shop said it will take 3 more days."


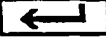
Screen description continued on the following page.

## Section 1 - Adjustor

On the Callback Menu, key X next to Adjustor for Tickets needing extensions. **ENTER**

ADJUSTOR CALLS			6 Print (Y or N) <b>(N)</b> CCCB10-1	
1 NEXT INSURANCE CO. _____				
2 (Bill to Name)	3 CUST#	4 (Phone Number)	5 # OF CALLS	
AMX* TESTING CUSTOMER*	AMXTEST	222-222-2222	002	
FIREMAN'S INSURANCE GROUP**	FIR0001	212-212-2121	003	
HANNOVER INSURANCE GROUP**	HAN0101	111-111-1111	001	
Cmd1=Exit Cmd7=AAI ROLL=Forward/Back Cmd24=Jump				

The Adjustor Screen lists all insurance companies alphabetically, including customer numbers, phone numbers and number of calls to be made to that office.


- Next Insurance Co. (optional) - Key your name. The name keyed appears at top of list. Press **ENTER**. The next list will appear. -OR-
  - Bill to Name - Key X by the Insurance Company Name you will need to call.  to move cursor down the list of names, as needed. Press **ENTER**. The Insurance Company Selection Screen will appear listing customers who need authorizations or extensions on rentals from that particular company.
- NOTE:** To view the Adjustor Selection Screen, key X by your name and **ENTER**. See example screen on following page.
- Customer Number - Displays insurance company "Bill-to Customer Number."
  - Phone Number - Displays adjustor phone number.
  - Number of Calls - Displays number of calls to be made to Adjustor(s).
  -  to move cursor to top right of screen. To Print a copy of the Adjustor Callback Screen, key Y=Yes, over default N=No.

**Reminder!** Use your roll-up and roll-down keys to scroll backward or forward through the Insurance Company listings.

## Section 1 - Adjustor Detail Screen

ADJUSTOR CALLBACK DETAIL - D100096		CB50-1
DATE OUT 06/17/94 8:19AM	CURR EXT 06/18/94	
CUSTOMER MURPHY* SHARON*	HOME PHONE# 333-333-3333	
RATE 15.00 /DAY	OFFICE PHONE# 333-343-3434 EXT 2323	
	OTHER PHONE# 222-222-2222	
BILL-TO Y	BILL-TO CUST# FIR0001	SHOP# ALLEN'S AUTO BODY
BILL-TO NAME FIREMAN'S INSURANCE GROUP	PHONE# 777-777-7777	ATTN BOB BROWN
PHONE# 212-212-2121 EXT 8896	YEAR MAKE/MODL	LOSS DATE 000000
ID/ATTENTION 001 SPOOL* LYNN	INSURED	
CLAIM/POL/PO# CLAIM 35		
MAX. AMOUNT		
NEW INFO EMP# <u>          </u>		
06/17/94 11:41 AM RENTAL EXTENDED UNTIL 06/13/94 BY LS 74024		
06/18/94 9:00 AM PROBLEM GETTING PART IN 75026		
STATUS IM DO X OK CALLBACK TYPE X B/S X ADJ SVC CUST		
EXTENSION DATE BY		
LAST DAY BY		
F2=Cust List F3=Exit F4=ID List F5=Open Ticket F6=Chgs To Date		
F7=AAI F8=Update Tkt F9=Callback Notes F10=Receipt Depos Roll=Fwd/Back		
F12=Previous F13=Insurance F14=Credit Check F15=Reservation		


The Adjustor Detail Screen is very similar to the Body Shop Detail Screen. Take a few moments to familiarize yourself with the above screen. From this screen you may change ticket information, call-back status, and extension date. Be sure to refer to the Body Shop Detail Screen, page 8-5, for field descriptions.

**NOTE:** To enter new information received from a current call,  to move cursor, key your employee number in the "EMP#" field highlighted above. Next, key all pertinent information in the message space provided, using the same abbreviations that were used in previous messages. Change the Status of the call from "DO" to "OK." Key 3 days from today as the Extension Date.

**This completes the review for Section 1 - Adjustor!**  
**Turn to the next page and continue the Callback Exercise for Section 1 - Customer.**



### Section 1 - Customer Cash/Check


The Customer option can also provide a list of customers who are paying by cash/check ONLY. To view this list from the customer calls, on the Callback Menu Screen, key X in both the "Customer" field and "Cash/Check Only" field **ENTER** .  to move cursor down the list of customer names, as needed. Key X to select the customer name to be displayed, press **ENTER** the Customer Detail Screen will appear.

### Section 1 - Customer Detail Screen

CUSTOMER CALLBACK DETAIL - D025116		CB50-1
DATE OUT 06/21/94	CURR EXT 06/20/94	
CUSTOMER DANIELS* KURT*	HOME PHONE# 999-888-4444	
RATE 32.99 /DAY	OFFICE PHONE# 999-777-9797 EXT 2222	
	OTHER PHONE# 222-222-2222	
BILL TO Y	SHOP#	
BILL TO NAME	PHONE#	
PHONE#	EXT	ATTN
ID/ATTENTION	YEAR	MAKE/MODL
CLAIM/POI/POI	LOSS DATE 000000	
MAX AMOUNT	INSURED	
NEW INFO	EMP# <input type="text"/>	
06/22/97 2:42 PM CUSTOMER EXTENSION DATE UNTIL 06/23/97 75757		
STATUS IM DO X OK	CALLBACK TYPE B/S ADJ SVC X CUST	
EXTENSION DATE 06/23/97		
F2=Cust List F3=Exit F4=ID List F5=Open Ticket F6=Chgs To Date		
F7=RAI F8=Update Tkt F9=Callback Notes F10=Receipt Depos Roll=Fwd/Back		
F12=Previous F13=Insurance F14=Credit Check F15=Reservation		

Any background information you need to know about the renter is available on this detail screen.

The Customer Detail Screen is very similar to the Body Shop Detail Screen. Again, take a few moments to familiarize yourself with the above screen. From this screen you may change ticket "bill-to" information, callback status, and customer extension date. To refresh your memory, refer to the Body Shop Detail Screen, page 8-5, for similar field descriptions.

**NOTE:** To enter *new* information received from a current call,  to move cursor, key your employee number in the "EMP#" field highlighted above. Next, key all pertinent information in the message space provided, using the same abbreviations that were used in previous messages. Change the Status of the call from "DO" to "OK." Key a new Customer Extension Date of 3 days from today and take any required deposits.


This completes the review for Section 1 - Customer!

Turn to the next page and continue the Callback Exercise for Section 1 - Service.


## Section 1 - Body Shop Selection Screen

SERVICE CALLS				11	CB07-1
				PRINT (Y OR N)	(N)
1		CALLBACK			
MONARCH CHEVY** 3		CURRENT STATUS			
987-654-3210 9999999		YEAR/MAKE/MODEL	TICKET#	GR/BR	EXT DATE SHOP ADJ
		4	5	6	7 8 9
2					
ABLE* ALICE*		94 FORD TAURUS	* D027415	0101	06/15/94 DO DO
		Notes			
		10			
DAVIS* TOM*		95 FORD MUSTANG	* D027419	0101	06/17/94 DO DO
		Notes			
JONES* SUE*		90 CHEVY LUMINA	* D027420	0101	06/20/94 DO DO
		Notes			

\* INDICATES TICKET WITH EXTENSION DATE GREATER THAN 2 DAYS OLD  
 Cmd1=Exit Cmd7=AAI Cmd12=Previous ROLL=Forward/Back

1. Displays Service Shop Name/Phone Number and 6-digit Customer Number.
2. Customer Name - Key X by your name.  to move cursor down the list of names, as needed. Press **ENTER**. The selected Service Shop Detail Screen will appear. See example screen on the following page.
3. Displays Customer Number.
4. Displays Year/Make/Model of customer vehicle.
5. Displays Rental Ticket Number. An extension flag (\*) indicates ticket has an extension date that is more than 2 days old. Definitely make all extension flags a priority!
6. Displays Group and Branch Numbers.
7. Current Ext. Date - Displays current extension of specific rental ticket.
8. Callback Status Shop - Displays shop callback status - LM (Left Message), DO (Make Call), OK (Call Made).
9. Callback Status Adjustor - Displays adjustor callback status - LM (Left Message), DO (Make Call), OK (Call Made).
10. Notes - Key Service Shop notes. Notes will forward to Service Shop Detail Screen.

**NOTE:** You must still go into each detail screen and update the callback status (LM/DO/OK) for each customer, even after keying notes.

11.  to move cursor to top right of screen. To Print a copy of the Service Shop Selection Screen, key Y=Yes, over default N=No.

**Reminder!** Use your roll-up and roll-down keys to scroll backward or forward through Shop listings.

## Section 1 - All

OPEN TICKETS						CB25-1	
1 NEXT CUSTOMER						7 PRINT (Y OR N) <b>(N)</b>	
2	3	4	5 CALLBACK TYPE			6	
	TICKET#	CURRENT EXT DATE	B/S	ADJ	SVC	CST	CUSTOMER EXT DATE
AKLSJDEKASJ* LASDF,L	D999089		-	-	-	-	
BAUCKE* CLAUDIA*	D999096	* 11/09/97	-	-	-	-	
BAUCKE* SVEN*	R111062	* 11/01/97	-	-	-	-	
BENOIT* TRICIA*	D999019		-	-	-	-	* 1/28/97
BOMBAY* BILL*	D999028		-	-	-	-	* 3/10/97
BOND* JAMES*	D999080	* 9/10/97	-	-	-	-	
BOOP* BETTY*	D999022		-	-	-	-	* 2/20/97

1 LEFT A MESSAGE  
\* INDICATES TICKET WITH EXTENSION DATE GREATER THAN 2 DAYS OLD

Cmd1=Exit Cmd7=PA1 ROLL=Forward/Back

The "All" Callback option provides a list of *all* Open Tickets, in alphabetical order, that need extensions.

- Next Customer - Key next Customer Name at which the Callback list should begin. Press **ENTER**. The list will appear. -OR-
- Customer Name - Displays name of rental customer.
- Ticket - Displays ticket Number. An extension flag (\*) indicates ticket has an extension date that is more than 2 days old. Definitely make all extension flags a priority!
- Current Ext. Date - Displays current extension date of specific rental ticket.
- Callback Type - Key X to select Callback Detail Record to display: B/S (Body Shop), ADJ (Adjustor), SVC (Service), and CST (Customer). A "#" by a Callback Type indicates a message has been left at that Callback Source (Body Shop, Adjustor, etc.). Press **ENTER**. The selected Callback Detail Screen will appear.
- Customer Ext Date - Displays customer extension date of specific rental ticket.
- ←** to move cursor to top right of screen. To print a copy of the Service Selection Screen, key Y=Yes, over default N=No.

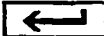
**Reminder!** Use your roll-up and roll-down keys scroll backward or forward through the Open Ticket listings.

Continue to the next page and learn how to select the "By Name" Callback option.

## Section 2 - Callbacks -or- Incomplete Reservations

RESERVATIONS INCOMPLETE FOR GPBR PPGM						CCRS11-B
8 Print (Y or N) <input checked="" type="radio"/> N						
1 Next Customer _____						
2 Customer Name	3 Date	4 Pickup Time	5 Stat	6 Car Type	7 Rent Type	
ASHBY* JOAN J*	6/15/94	2:00 PM	DEL	MINI	I	
EARP* WAYNE*	5/16/94	8:00 AM	W/IN	INTM	D	
EASTWOOD* BILL*	5/10/94	4:00 PM				
GARCIA* MARIA*	6/19/94	2:00 PM	W/IN			
GORDAN* JILL*	5/01/94	1:00 PM				
GRAND* TIM*	6/20/94	10:00 AM	DEL	MINI	D	
HUME* BILL*	7/02/94	1:00 PM	W/IN	STD	B	
KING* MARGE*	5/18/94	5:00 PM	P/U	MINI	D	
LEE* DAVID*	7/13/94	3:30 PM	W/IN			
MONROE* JOE*	6/25/94	8:00 AM	CWC	SPEC	I	
NOWARK* BOB*	5/02/94	2:30 PM	P/U	FULL	R	
+-----+						
Cmd1=Exit Cmd3=Main Menu Cmd7=AAI Cmd12=Previous Screen Roll=Fwd/Back						

The Callback-Incomplete Reservations Screen alphabetically lists all reservations marked as incomplete when the reservation was originally opened. Reservations are selected from this list to make contact with the customer and complete the reservation, as needed.

1. Next Customer - Key next Customer Name at which the Callback list should begin. Press **ENTER**. The list will appear.
2. Customer Name - Displays name of rental customer.
3. Date - Displays pickup date of rental vehicle.
4. Pickup Time - Displays pickup time of rental vehicle.
5. Status - Displays rental status - W/IN (Walk In), DEL (Delivery), P/U (Pickup), CWC (Customer Will Call).
6. Car Type - Displays rental vehicle size.
7. Rental Type - Displays rental category - I (Insurance), B (Body Shop), D (Dealership), R (Regular), C (Corporate), O (Other).
8.  to move cursor to top right of screen. To print a copy of the Incomplete Reservations Screen, key Y=Yes, over default N=No.

**Reminder!** Use your roll-up and roll-down keys to scroll backward or forward through the Incomplete Reservation listings.

**Turn to the next page and learn about the Callback - No Show Reservations!**

**FUNCTION KEYS FOR OPTION 12 - CALLBACKS**

---

Listed on the bottom portion of each screen are several Function (F) keys. Each one performs a function that may be used while creating or viewing Callbacks.

<b><u>F2 = Cust List</u></b>	Displays a list of customer names and numbers that are to be used for billing purposes.
<b><u>F3 = Exit or Main Menu</u></b>	Allows you to exit the program or return to the Main Menu Screen.
<b><u>F4 = ID List</u></b>	Displays a list of ID numbers for individual adjustors, agents, etc. for each customer number.
<b><u>F5 = Open Ticket</u></b>	Displays the Open Rental Ticket for any rental background information.
<b><u>F6 = Chgs To Date</u></b>	Displays the total number of rental days, total dollar amount, less deposits, and balance owed.
<b><u>F7 = AAI</u></b>	Allows access to the Inquiry Programs.
<b><u>F8 = Update Tkt</u></b>	Allows you to update Open Rental Ticket information such as claim numbers, shop name, etc., as needed.
<b><u>F9 = Callback Notes</u></b>	Allows you to key additional information.
<b><u>F10 = Receipt Depos</u></b>	Allows you to update payments on Open Rental contracts.
<b><u>F12 = Previous</u></b>	Return to previous screen.
<b><u>F13 = Insurance</u></b>	Allows you to key renter insurance information.
<b><u>F14 = Credit Check</u></b>	Allows you to key credit check information.
<b><u>F15 = Reservation</u></b>	Allows you to continue to the next reservation.

***Congratulations!***

**You have successfully completed the entire Callback Exercise. Now, you are ready to update live Open Rental Tickets at your Office!**

## EXERCISE 9 CLOSING A TICKET

This exercise will guide you through Option 5 - Closing a Ticket. Close the ticket from Exercise 2 - Open a Ticket, Calendar Day.

Key Option # 5 and the Daily Rental Ticket Number on the ECARS Training Menu. **ENTER** The following screen will appear.

Screen 1

1. CLOSING TICKET# 999010 For Renter SMITH, ROBERT	
Charges End On Date 062494 Time 1102 AM	
2. Mileage	
Unit P00326 Start End Unit Start End	
Unit Start End Unit Start End	
Fuel Charges \$ 5.00 Per RENTAL	Amount To Be Charged .00
Last Unit Returned Here Or To	Drop Charge .00
Misc Charges/Refundable Expenses (X if applicable)	SURCHG 2.05 /DAY
3. Bill To (Y/N) N Cust# Name	
Address	
City ST Zip	
Curr Ext 06/24/94 per INC	ID/Attn
4. Billing Dates (If Diff) From Date 000000 Time 0000	
Max. Amt. To Date 000000 Time 0000	
5. Portion To Be Billed (Select One)	
1. Total Charges Less DW PAI Fuel Tax/Schg Mileage Drop	
Misc Airport Access	
2. \$ .00 Per Day Plus Tax/Schg (Not To Exceed \$ .00 Per Day)	
3. \$ Per Day No Tax/Schg (Not To Exceed \$ .00 Total)	
4. Other Amount \$ .00 Description	
Cmd2=Cst List	
Cmd4=ID List Cmd5=Callback Detail Cmd7=AAI Cmd8=Accident Rpt Cmd9=Close Pending	

There are 5 basic sections to Screen 1.

- 1. Closing Ticket#:** This section displays the DR Ticket Number, Renter Name and the Date and Time the charges end. The current date and time are provided by the computer, but can be changed by keying over date shown.
- 2. Mileage:** This section requires the Starting and Ending mileage for each unit, along with Fuel Charge, Misc. Charges, etc.
- 3. Bill To:** This section allows you to key or delete billing information as necessary. Also displays Current Extension Date from callbacks.
- 4. Billing Dates (If different):** This section allows you to key billing dates if they are different from the actual dates of the contract.

## Section 1 - Closing Ticket #

1	
CLOSING TICKET#	999010 For Renter SMITH* ROBERT*
2 Charges End On	Date 062494 Time 1102 AM

1. Verify the Ticket # and Renter Name.
2. The length of this rental is determined by the current extension date. If necessary, adjust the date by keying directly over the information displayed.

**NOTE:** If ticket is close pended, this date may not be changed. You will need to unpend the ticket in order to change the date.

## Section 2 - Mileage

Mileage		1	
Unit #00326	Start 2500	End 2599	Unit
Unit	Start	End	Unit
2			
Fuel Charges @ 7.00 Per RENTAL	Amount To Be Charged		00
Last Unit Returned Here	Or To 3	Drop Charge	5.00
Misc Charges/Refundable Expenses	(X if applicable)	SURCHG	.00 /DAY

1. Key the Starting and Ending Mileage for each unit.
2. Key the Fuel Charge as designated by your Group. Each Group will have a standard fuel charge.
3. This vehicle has been "dropped" at another Group Branch. Key in the Group/Branch Number.

**NOTE:** You may see an error message requesting the mileage be confirmed. Because this is a training program, you may key an "X" in the selection field for mileage confirmation without updating any files. For everyday branch use - you must verify the mileage you keyed is correct before confirming that mileage.

## Section 3 - Bill To

Bill To (Y/N) Y	Cust# 999999	Name	YOUR NAME**		
Address		555 SOMEWHERE LANE			
City		Anytown	ST	MO	Zip 12345
ID/Attn		999 CHARLIE BROWN			
CURR Ext. 06/24/94 per LNC.					

Verify the billing information. This information may be updated/changed/deleted if necessary.

## Screen 2

ACCEPT (A) or RECOMPUTE (R) _ 1				Cmd1=Exit Cmd7=AAI					
2	3	4	5	6	7	8	9	10	
	Miles	Hours	Days	Weeks	Months	Waiver	PAI	Special	
062094									
0315 PM	#		3			3	3		
062394	@		16.99			7.00	1.00		
1120 AM	\$		50.97			21.00	1.00		
000000									
0000	#								
000000	@								
	\$								
000000									
0000	#								
000000	@								
	\$								
000000									
0000	#								
000000	@								
	\$								
TOTALS		.00	.00	50.97	.00	.00	21.00	3.00	.00
11	Surcharge % 5.850		Gas 12 7.00	Disc @ 14 0 %		16		TOTAL 89.95	
	Surcharge 2.98		Drop 13 5.00	Misc 15 .00		Cmd3=Restart			

Screen 2 breaks down all charges and displays the grand total. Use this screen to write down all charges on the rental ticket when the customer returns the vehicle.

1. Accept or Recompute (when adjustments are needed).
2. Four rate sections. Displays beginning and ending date/time of each rate change.
3. Number of Miles, rate per mile, total amount of overmileage.
4. Number of Hours, rate per hour, total hourly charges.
5. Number of Days, rate per day, total daily charges.
6. Number of Weeks, rate per week, total weekly charges.
7. Number of Months, rate per month, total monthly charges.
8. Number of Days, rate per day, total DW charges.
9. Number of Days, rate per day, total PAI charges.
10. Number of Days, rate, total Special rate.
11. Sales Tax rate and total.
12. Fuel Charge.
13. Drop Charge.
14. Discount percentage and total.
15. Miscellaneous Charges.
16. Grand Total.

Key "A" **ENTER** to accept the charges and advance to Screen 3.

<b>CMD 1</b>	Exit to Enter Request Prompt.
<b>CMD 3</b>	Restart.
<b>CMD 7</b>	AAI. Access Inquiry Programs.



## Closing Procedure - Calendar Day

## Screen 3 - continued

CLOSING TICKET # 999011 For Renter SMITH* ROBERT*			
Total Charges	89.95		
Less Deposits	.00	0 # Of Deposits	
Less Amount Billed To	89.95	Your Name**	
Balance Due	10.05	Refund	

Settlement Of Balance Due

Paid	<.00	By Cash	
	.00	By Check	
	.00	By Credit Card	
Bill	.00	To Cust#	A/R Description
Bill	.00	To Cust#	A/R Description
Bill	.00	Name:	(Last*First*) TTL
		Attn:	
		Street	
		City	ST Zip
		Home Phone 000 000 0000	Office Phone 000 000 0000
		A/R Description	
Charge	.00	To Account	Unit # Desc
Charge	.00	To Account	Unit # Desc
Refund	10.05	By Cash X Or Check	
Emp#	01053		

Cmd1=Exit Cmd2=Cust List Cmd5=Internal Accts Cmd8=Deposits Cmd10=CC Apprvl  
 Cmd3=Restart Cmd4=ID List Cmd6=Callbk Detail Cmd9=ClosePnd Cmd11=CK Apprvl

The total charges for this rental ticket are shown. There is also an amount that is to be billed to a third party - Your Name. An amount due may or may not be shown.

If there is an amount due, key the amount in the field before "By Cash." Key your 5-digit Employee Number.

If there is a refund due, key the refund amount next to "Refund" and X cash. Key your 5-digit Employee Number.

**ENTER** to accept the information and close the ticket. RALPH will receipt the cash you accepted from the customer and bill the third party, Your Name, for their amount.

Now, let's "unpend" the same 24 hour ticket previously pended.

On the ECARS Training Menu Screen, key Option 5 and the Ticket Number **ENTER**.  
Again, Screen 1 will appear.

### Section 1 - Closing Ticket #

CLOSING TICKET# 060994 For Renter SMITH* ROBERT*			
Charges End On Date 061094 Time 0950 AM			
Mileage			
Unit PP0027	Start 0	End 150	Unit Start End
Unit	Start	End	Unit Start End
Fuel Charges @ 7.00 Per RENTAL	Amount To Be Charged .00		
Last Unit Returned Here X Or To	Drop Charge .00		
Misc Charges/Refundable Expenses (X if applicable)			
Bill To (Y/N) N Cust#	Name		
	Address		
	City	St	Zip
Curr Ext 091194 per	ID/Attn		
Billing Dates (If Diff)	From Date 082294	Time 0937 AM	
Max Amt	To Date 091194	Time 0950 AM	
UN-PEND TICKET			
Press Enter to confirm that you wish to Un-Pend this Daily Rental Ticket.			
Cmd1=Exit Cmd12=Previous			Emp#
Cmd2=Cst List			
Cmd4=ID List Cmd6=Callback Detail Cmd7=AAI Cmd8=Accident Rpt			
Cmd9=Un-Pend			

Press **CMD 9** to "Unpend" the ticket.

Key your Employee Number in the Emp# field. Press **ENTER** to confirm that you wish to unpend the ticket. The ECARS Training Menu Screen will return.

*Turn to the following page to close the 24 hour ticket as normal.*

## Screen 2

ACCEPT (A) or RECOMPUTE (R) <u>A</u>		Cmd1=Exit Cmd7=AAI						
	Miles	Hours	Days	Weeks	Months	Waiver	PAI	Special
062394			2			2	2	
0842 AM #								
062494 @	Rate 1		24.95			9.00	1.00	
0800 AM \$			49.90			18.00	2.00	
062494			1			1	1	
0800 AM #								
062594 @	Rate 2		32.99			9.00	1.00	
1004 AM \$			32.99			9.00	1.00	
000000								
0000 #								
000000 @								
\$								
000000								
0000 #								
000000 @								
\$								
TOTALS	.00	.00	82.89	.00	.00	27.00	3.00	.00
								TOTAL
Surcharge @	5.850	Gas	7.00	Disc @	0 @			
Surcharge	4.85	Drop	.00	Misc	.00	Cmd3=Restart		
								121.74

Screen 2 breaks down each rate charged on the ticket.

Key an "A" to accept the charges. Write down the charges on the rental ticket if available.

**ENTER** to advance to Screen 3.

**CMD 1** Exit to Main Menu.

**CMD 3** Restart.

**CMD 7** AAI. Access Inquiry Programs.

## Check Refund Request

## CHECK REFUND REQUEST

RENTER: SMITH\* ROBERT\*  
123 MAIN STREET  
ANYTOWN MO 63124

Amount: 78.26

Pay To:  
Cust # 999999  
1 Name SMITH\* ROBERT\*  
Address  
123 MAIN STREET  
ANYTOWN MO 63124 0000

2 Reason overpayment - \$200.00 deposit.

3 Special Instructions  
Hold Check - Customer will pick up.

Cmd1=Exit Cmd7=AAI

The name and address of the renter will be displayed along with the amount of the refund.

1. If the check should be made payable to another party, key the information in the Pay To section.
2. Key the Reason for the refund (overpaid, etc.).
3. Key any Special Instructions (customer will pick up check, etc.).

**ENTER** to accept the information and close the ticket.

**CMD 1** Exit to Closing Ticket Screen 3.

**CMD 7** AAI. Access Inquiry Programs.

## **EXERCISE 10**      **CASH MANAGEMENT**

The Cash Management Option - Option 15 on the ECARS Training Menu - provides important information regarding Credit Card and Check approval procedures. It also includes a cash summary option that will allow you to balance the summary created while you opened and closed rental tickets in the previous exercises.

On the ECARS Training Menu, key Option 15 **ENTER**. The following screen will appear.

CCRT15-01

CASH MANAGEMENT

1. Credit Card Approval
2. Check Approval
3. Cash Summary

Option \_\_\_\_\_

Cmd1=Exit    Cmd7=AAI    Cmd12=Previous

Key Option 1 - Credit Card Approval **ENTER**. Carefully read the two screens displaying credit card approval "Things to Know".

Do the same for Option 2 - Check Approval. Become familiar with the **CMD 10** = CC Approval and **CMD 11** = CK Approval windows, then move on to Option 3 - Cash Summary.

**CMD 1**    Exit to Main Menu.

**CMD 7**    AAI. Access Inquiry Programs.

**CMD 12**    Previous Screen.

**CASH & CHECK SUMMARY - Selection 1, CASH/CHECK REVIEW**

1. Key your Employee Number.
2. Key an "X" in the Cash/Check Review selection field. **ENTER** The following screen will appear.

1	2	3	4	5	6	7	8	9
Line	Name	Cust-#	Doc-#	BRN	Credit	Credit	Other	Other
					2200 \$	2210	Acnt	Amount
0001	SMITH ROBE	999999	D999010	GM	100.00 \$	.00	0000	.00
0002	SMITH ROBE	999999	D999012	GM	50.00	.00	0000	.00
0003	SMITH ROBE	999999	D999011	GM	33.99 \$	.00	0000	.00
					.00	.00	0000	.00
					.00	.00	0000	.00
SUMMARY: Cash			10	133.99	Checks		11	50.00
						TOTAL		12
								13
Cmd1-Exit			Cmd7-AAI		GPBR PGM		Date 11/24/92	
					13		14	
							Line	
							15	

This screen displays all cash and check transactions. Cash transactions are noted with a cash symbol (\$) to the right of the dollar amount.

1. Line Number.
2. Customer Name.
3. Customer Number.
4. Document or Unit Number.
5. Branch #.
6. Credit Account 2200 Amount.
7. Credit Account 2210 Amount.
8. Other Account Number (any Account Number other than 2200 or 2210).
9. Other Amount (corresponds to Other Account Number keyed in field #8).
10. Total Cash.
11. Total Checks.
12. Cash/Check Summary Total.
13. Gpbr Number.
14. Summary Date.
15. Line #. Key a line number in this field to automatically roll screen to that line.

**ENTER** to return to the Cash Summary Menu.

**CMD 1** Exit to Cash Summary Menu.

**CMD 7** AAI. Access Inquiry Programs.

**CASH & CHECK SUMMARY - Selection 3, CASH/CHECK DEPOSIT**

		Employee #	01053
		1	
***** CASH & CHECK SUMMARY *****			
1.	Cash/Check Review		
2.	Make Adjustment Line		
<input checked="" type="checkbox"/> 3.	Cash/Check Deposit		
	Total Cash in Box	233.99	3
	Assigned Cash Box Amount	100.00	4
	Chits to be Reimbursed		5
	Chits to Remain in Box		6
	Total Checks to Deposit	50.00	7
***** CREDIT CARD SUMMARY *****			
1.	Credit Card Review		
		Customer #	
Cmd1=Exit Cmd 6=Petty Cash Cmd7=AAI Help			

1. Key your Employee Number.
2. Key an "X" in the Cash/Check Deposit selection field.
3. Key Total Cash Box amount. This is the total amount of cash in your box, including the assigned cash box amount. Assigned cash box amounts vary from Group to Group. For this exercise, use the "Cash" total from the Cash/Check Review plus \$100.00.
4. Key Assigned Cash Box amount - \$100.00.
5. Chits to Reimburse. There are no chits for this exercise.
6. Chits to Remain in the Cash Box. There are no chits for this exercise.
7. Key the Total Checks to be Deposited. For this exercise, use the "Checks" total from the Cash/Check Review.

**ENTER** The screen will display a message verifying the cash and checks balanced and a copy of the Cash Summary will print from the Plainpaper Printer.

Cash Balanced  
Checks Balanced

There are several other messages that may be displayed when balancing the cash summary. For a full display of error messages, press the **HELP** key.

**Congratulations!**  
**You have now successfully completed the ECARS Workbook!**

If you feel uncomfortable with any of the options, or just feel you would like a little more practice, you may open and close tickets, switch units, change rates, etc. in the ECARS Training System until you feel you are ready to go "LIVE".

The following list of Error Messages are the most commonly seen while completing the ECARS Workbook. You may refer to them any time you need a more detailed explanation than is displayed on the screen.

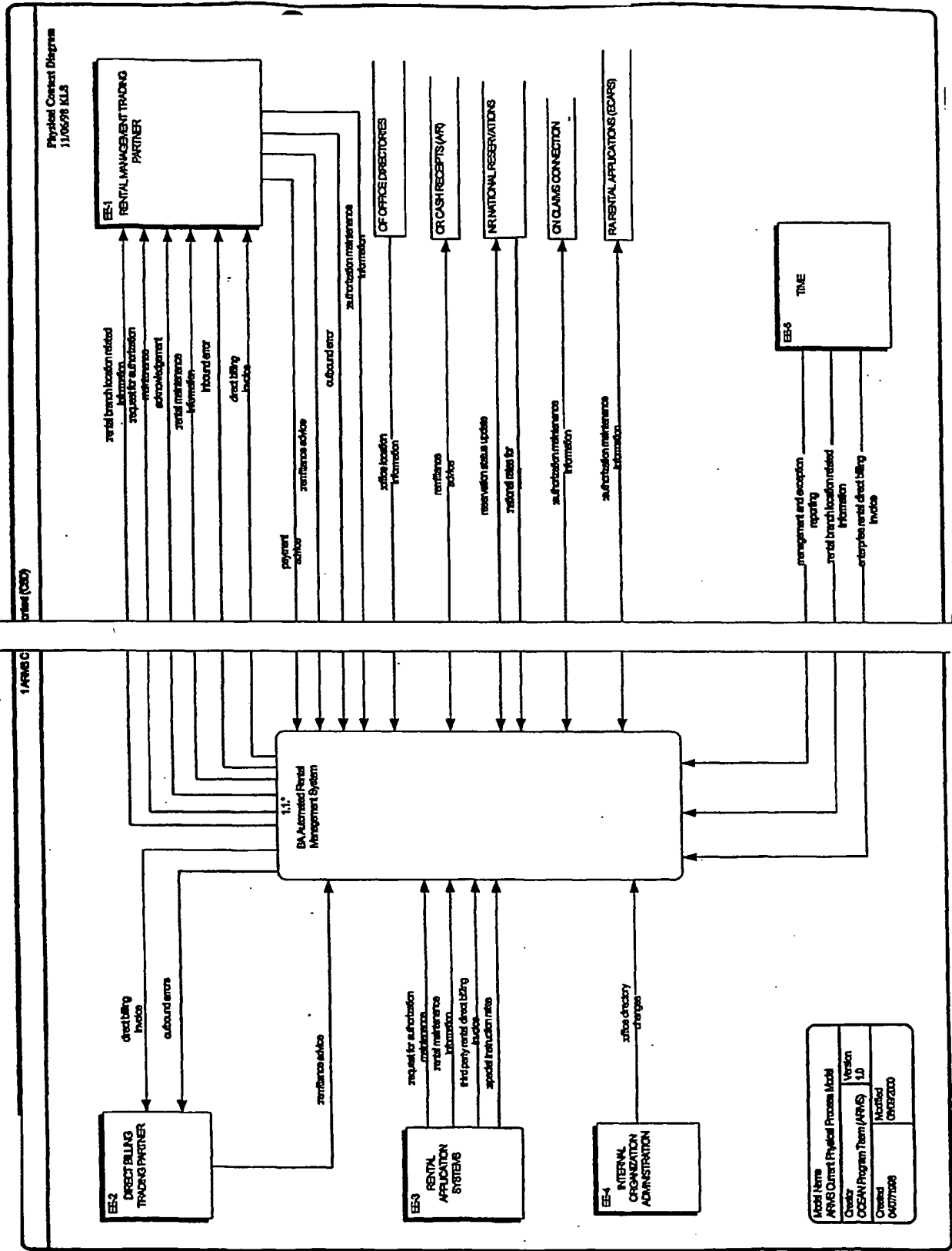
<b>Accept or Recompute?:</b>	Key A-accept or R-recompute to accept charges as they are, or make changes and recompute.
<b>Additional Driver?:</b>	The field for additional driver has been left blank. Key Y or N.
<b>Additional Driver Name Required:</b>	The field for additional driver has been marked Y, but no name has been supplied.
<b>Charges Do Not Balance:</b>	The payments that have been keyed do not balance to the amount due. Only Balance Due amount should be receipted.
<b>Confirm Date/Time Out:</b>	Date and Time keyed is not the current date and time. RALPH needs confirmation.
<b>Confirm Unit To Be Rented:</b>	Unit rented on another ticket or may be owned by another Branch in your Group. Always has a message explaining the problem.
<b>Date Out Invalid:</b>	Date out has been keyed incorrectly, or field left blank.
<b>Rental Type Invalid:</b>	Rental Type field has been left blank, or rental type keyed is not valid.
<b>Renter Name Invalid, Last*First*:</b>	Renter name has not been constructed correctly.
<b>Time Out Invalid:</b>	Time out has been keyed incorrectly, or field left blank.



EXHIBIT B

Context Rental and ARMS

1.  
ARMS Context  
(CSD)



1.1 BA Automated Rental Management System

Application Areas  
07/13/98

1.1.1.\*  
AA Trading Partner  
Business  
Transactions

1.1.2.\*  
AA Rental Systems  
Business Transactions

1.1.3.\*  
AA Office Information  
Synchronization Between  
Enterprise and Trading  
Partner

1.1.4.\*  
AA ARMS Support

Model Name ARMS Current Physical Process Model	
Creator OCEAN Program Team (ARMS)	Version 1.0
Created 04/07/1998	Modified 08/09/2000

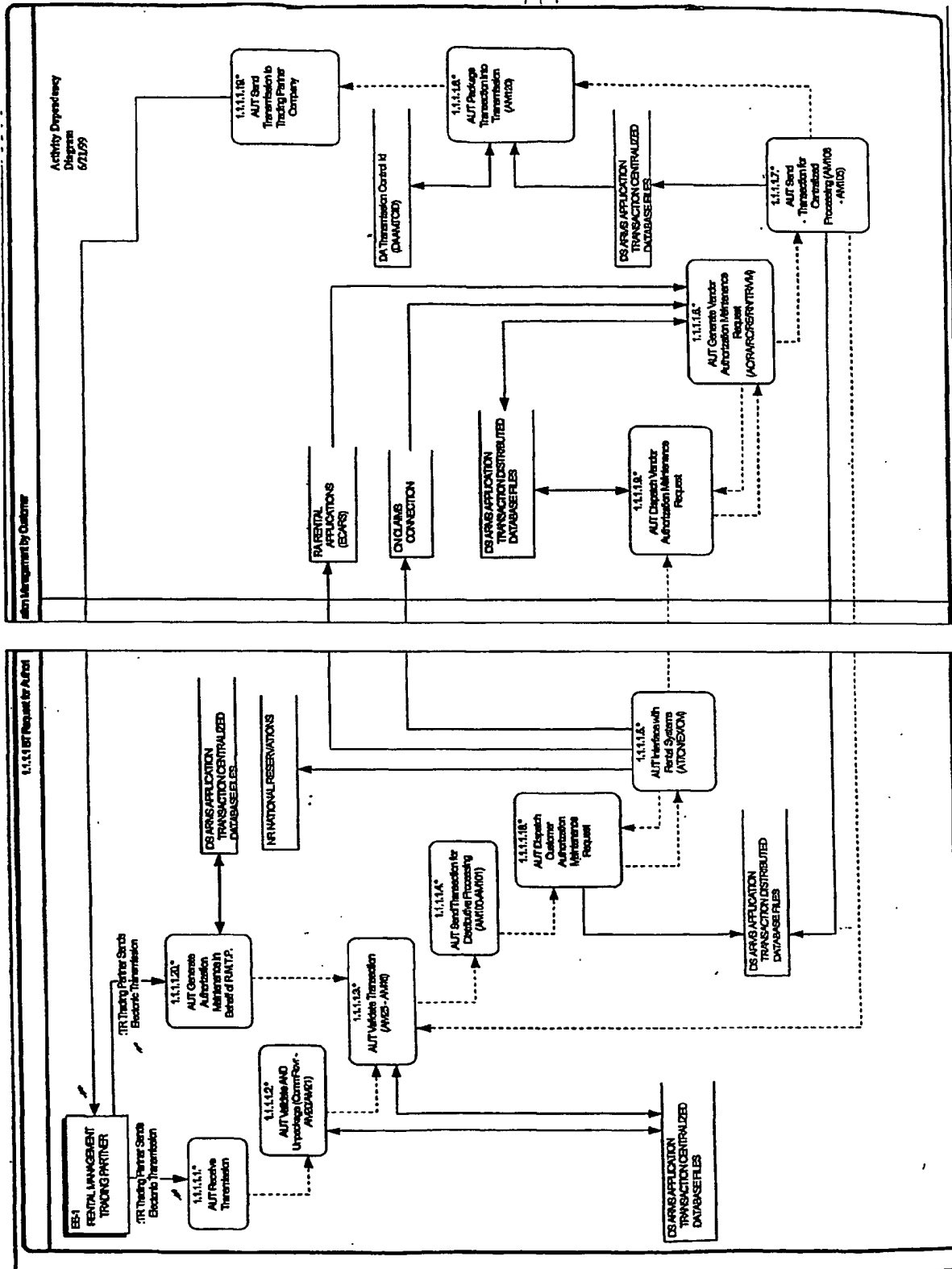
1.1.1 AA Trading Partner Business Transactions

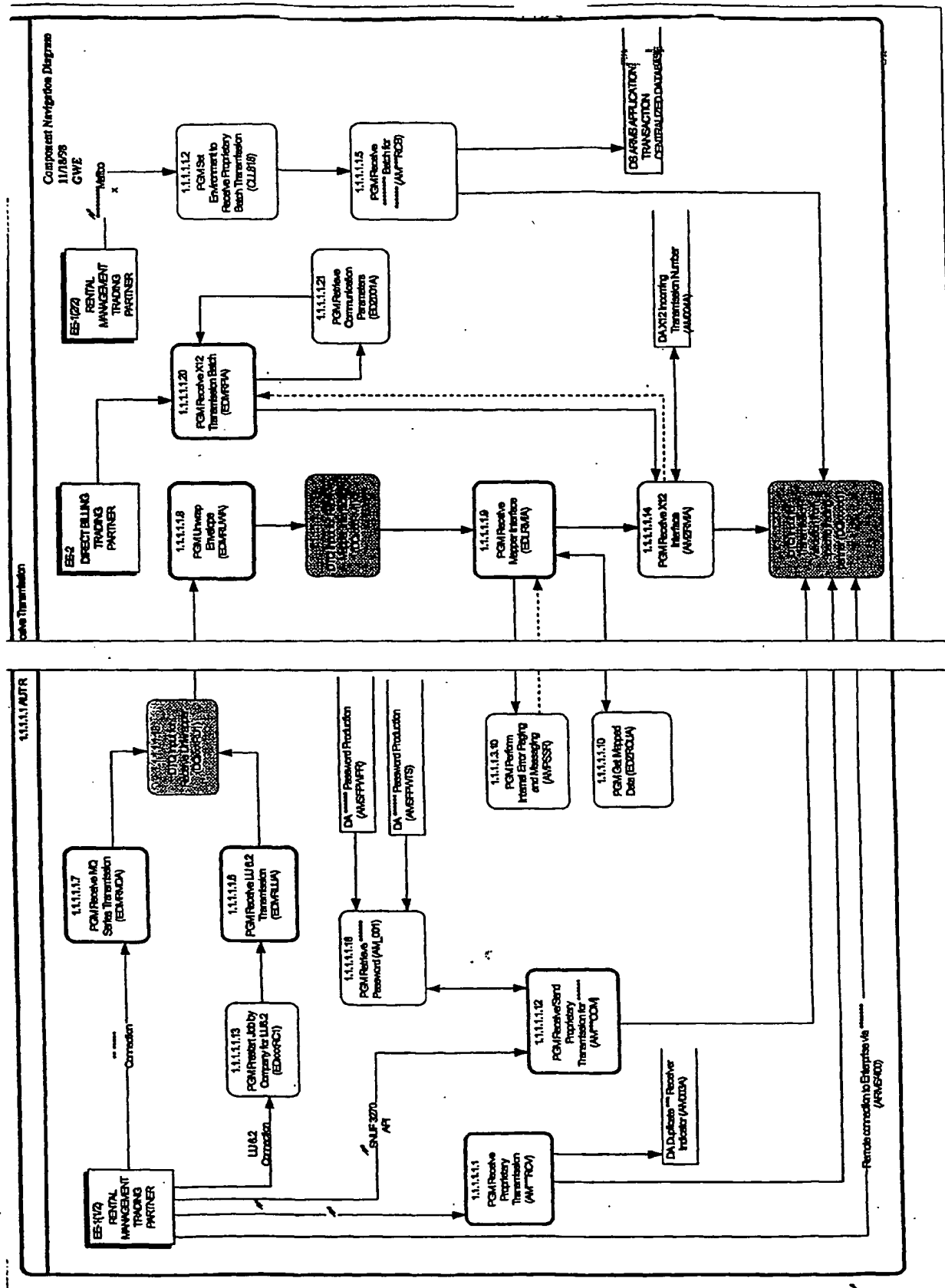
Business Transactions  
07/13/98

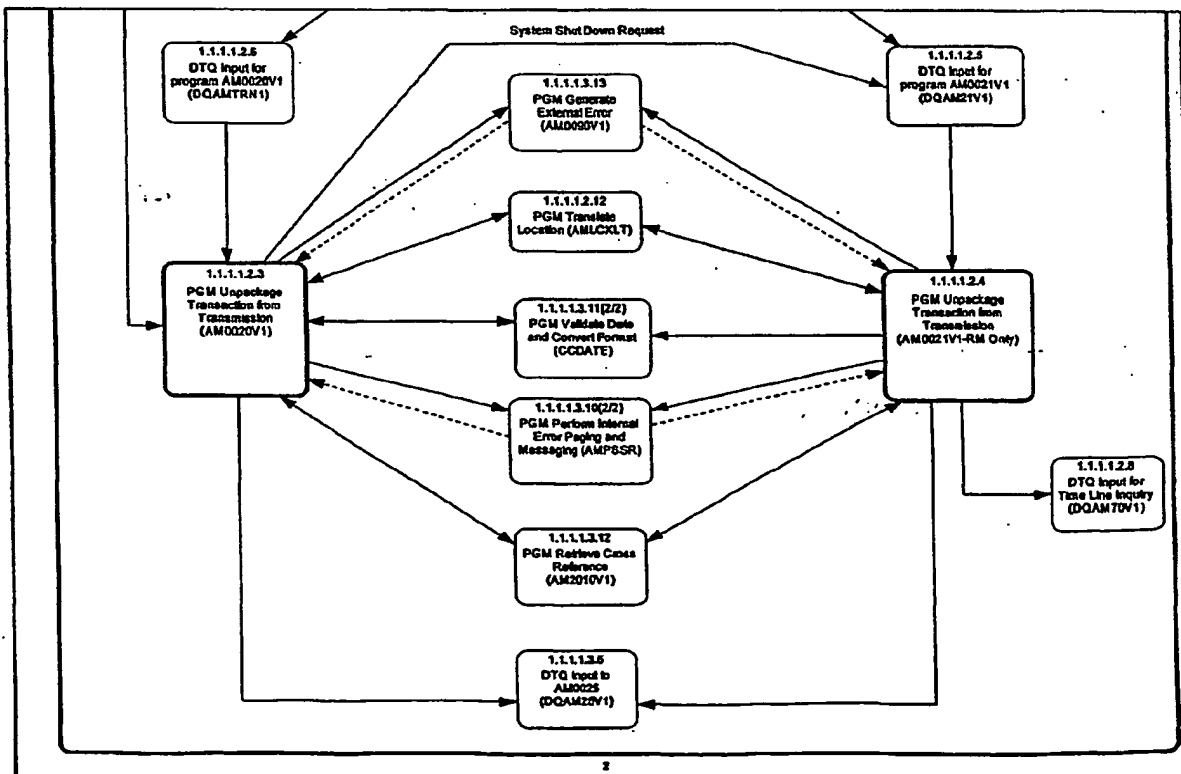
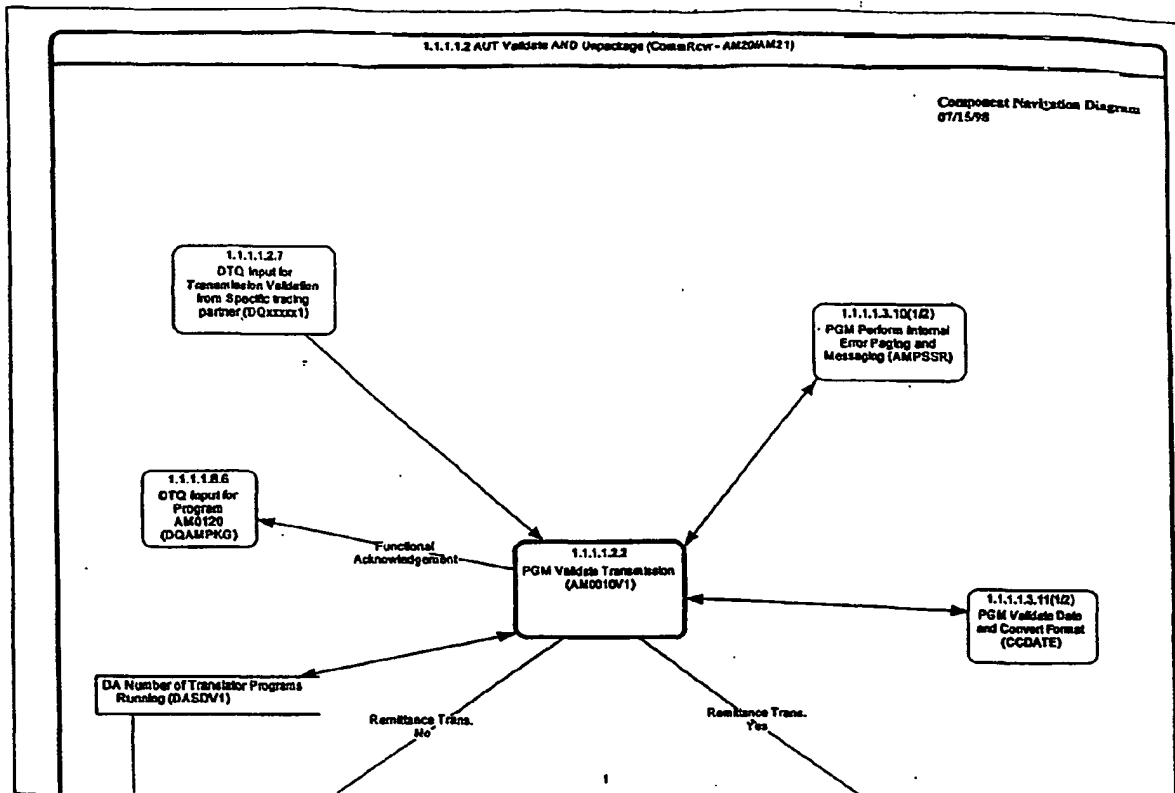
1.1.1.1.\*  
BT Request for  
Authorization  
Management by  
Customer

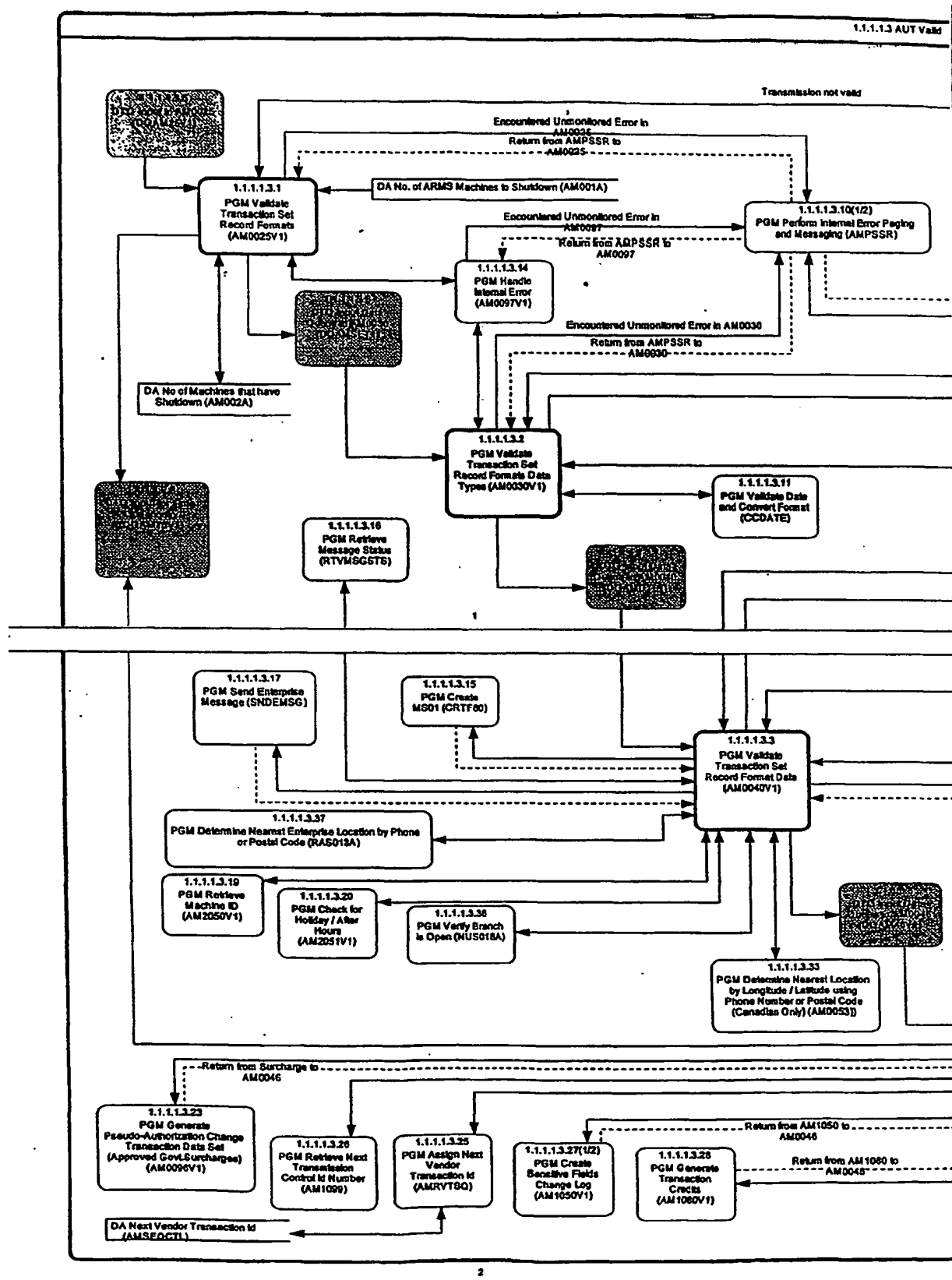
1.1.1.2.\*  
BT Manage Customer  
Remittance

1.1.1.3.\*  
BT Manage Customer  
Message

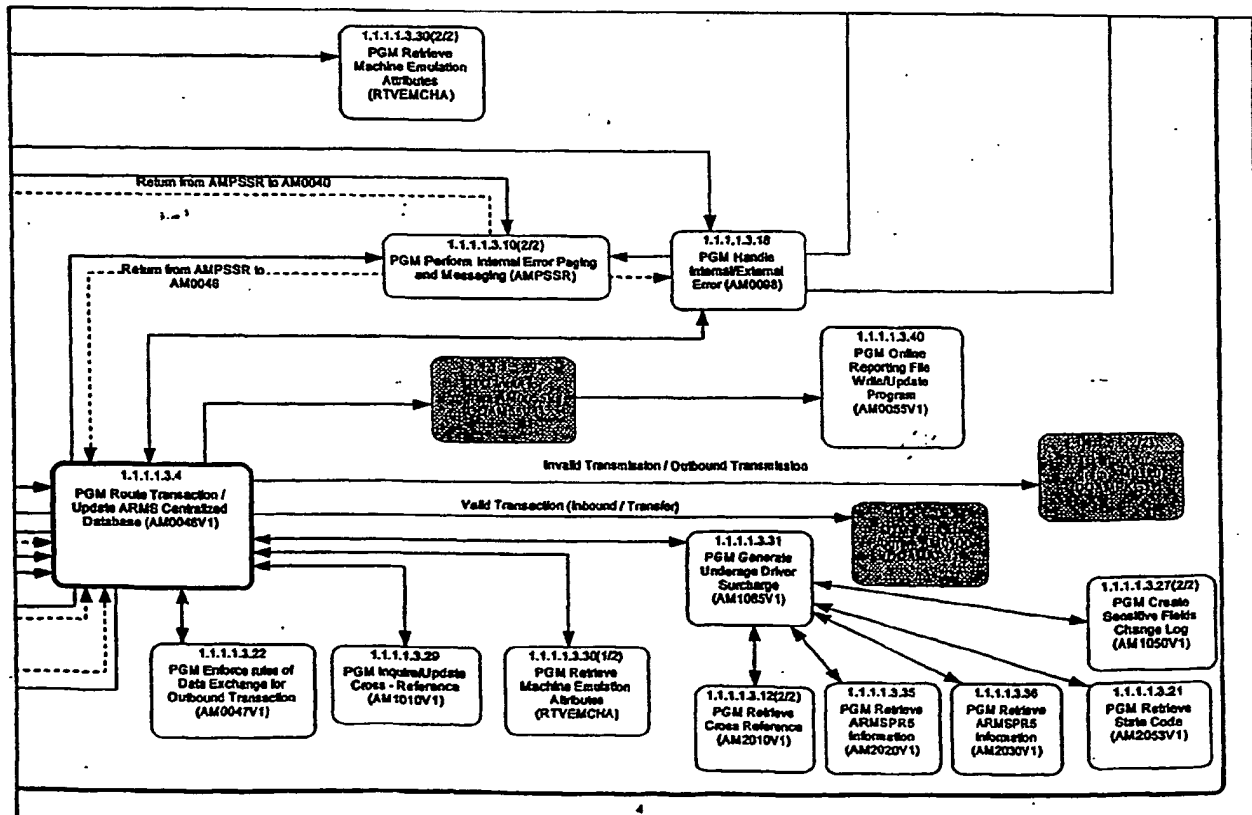
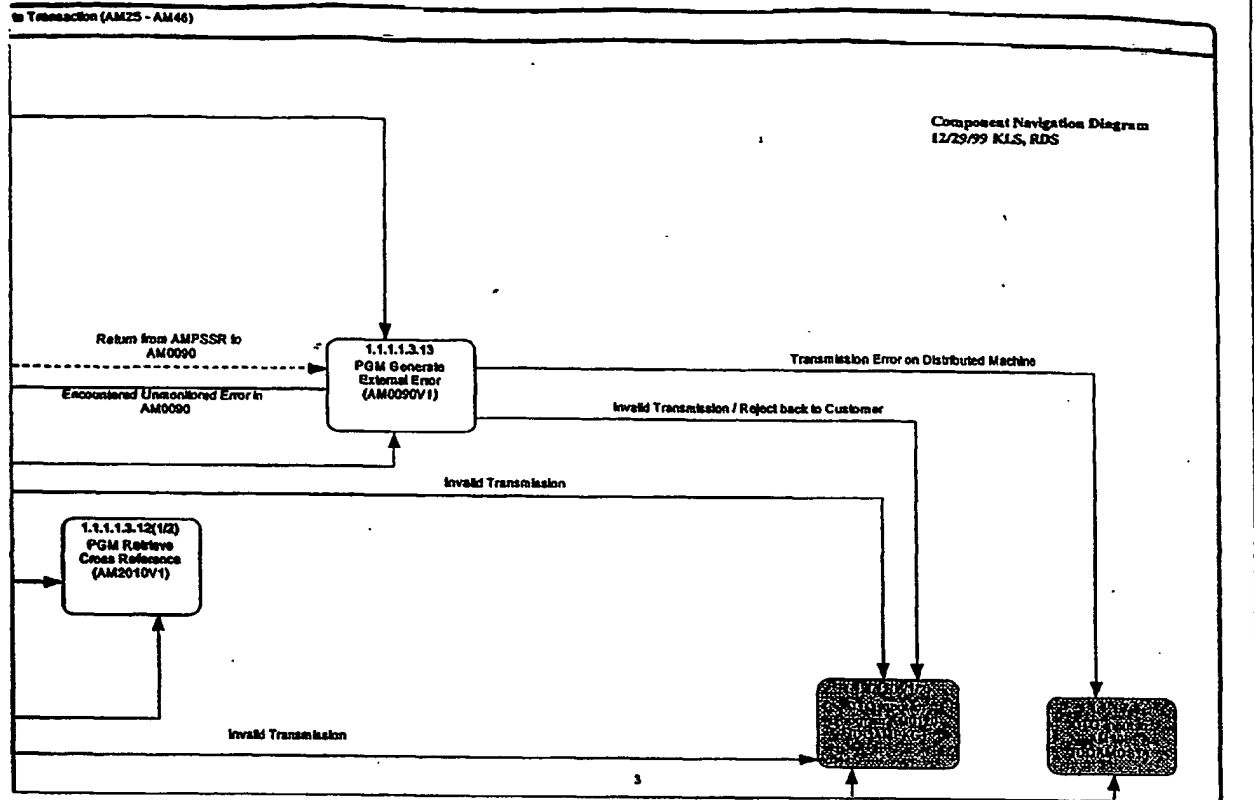




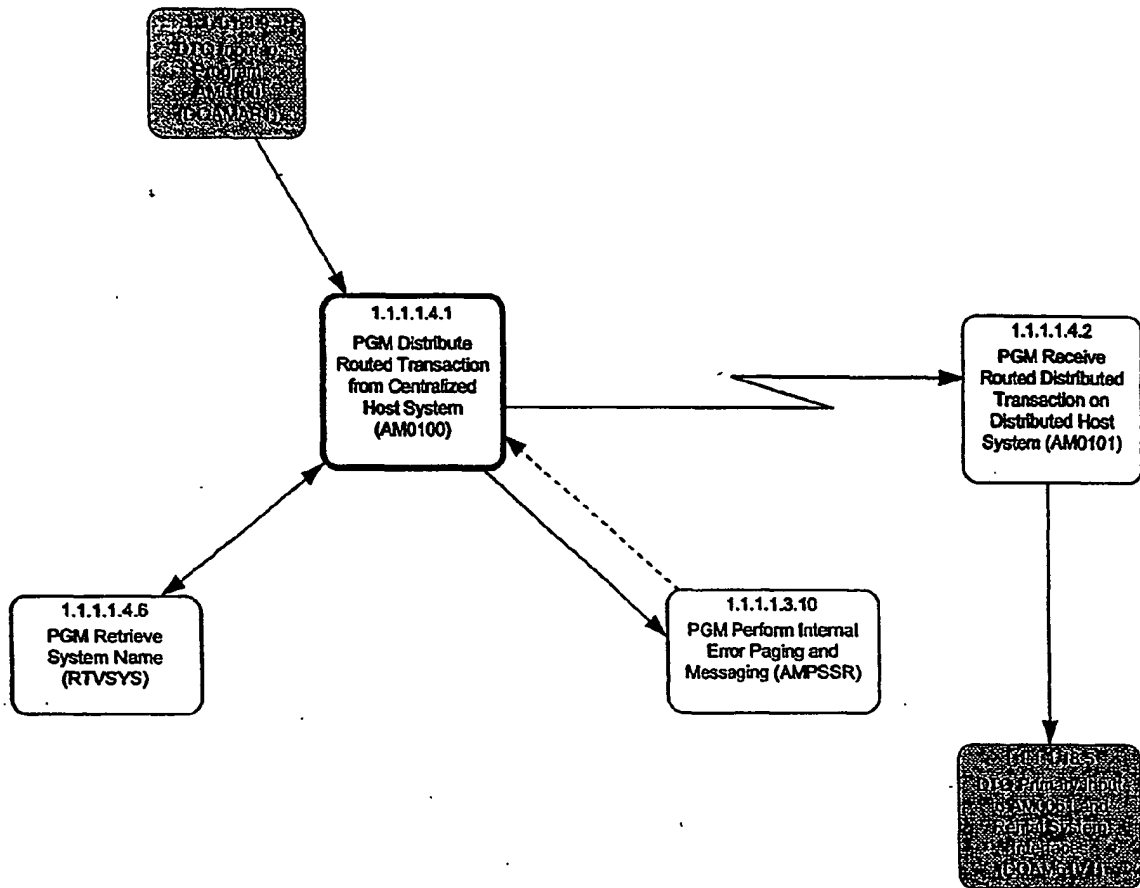








## 1.1.1.1.4 A/JT Send Transaction for Distributive Processing (AM100-AM101)

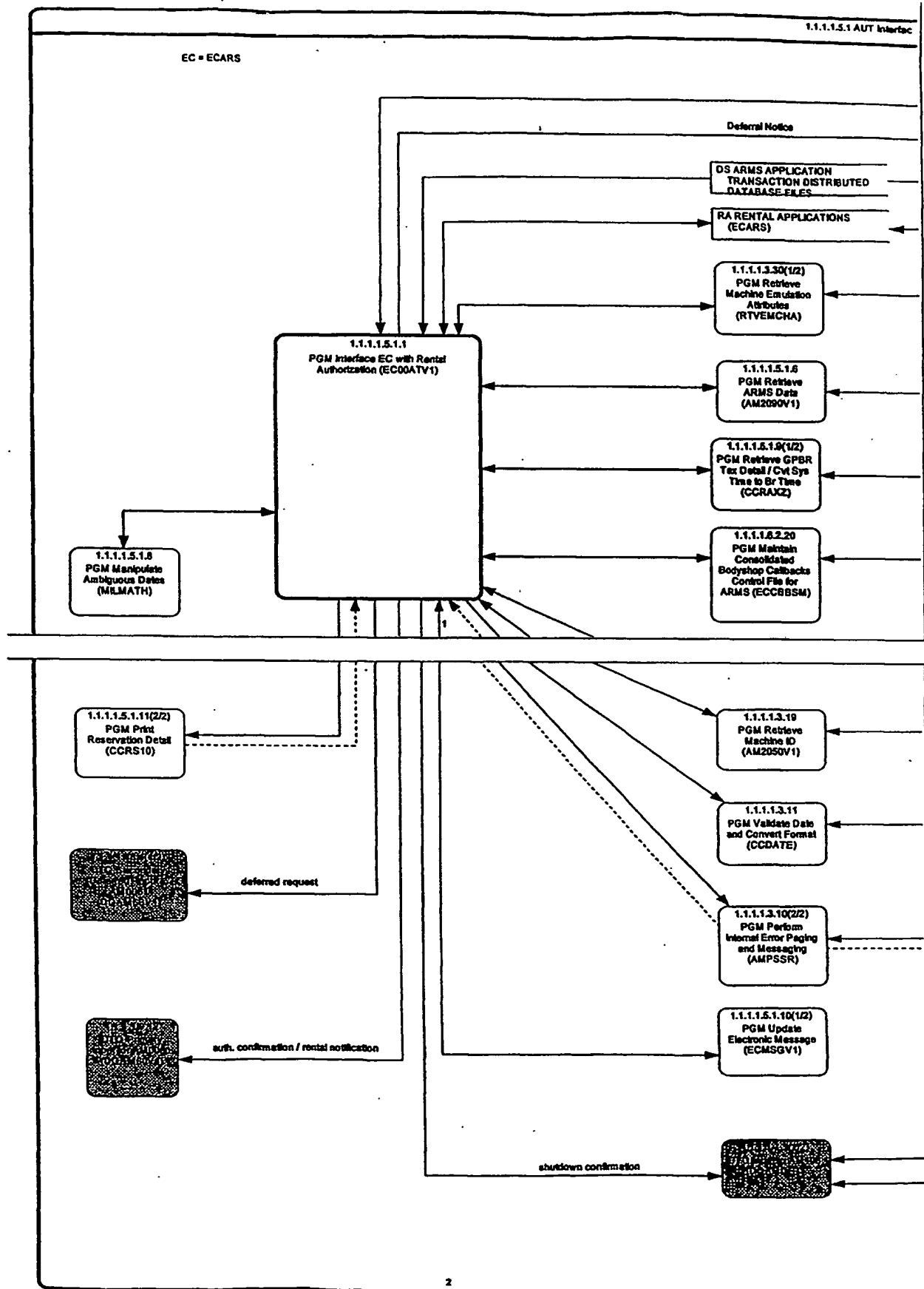
Component Navigation Diagram  
07/15/98

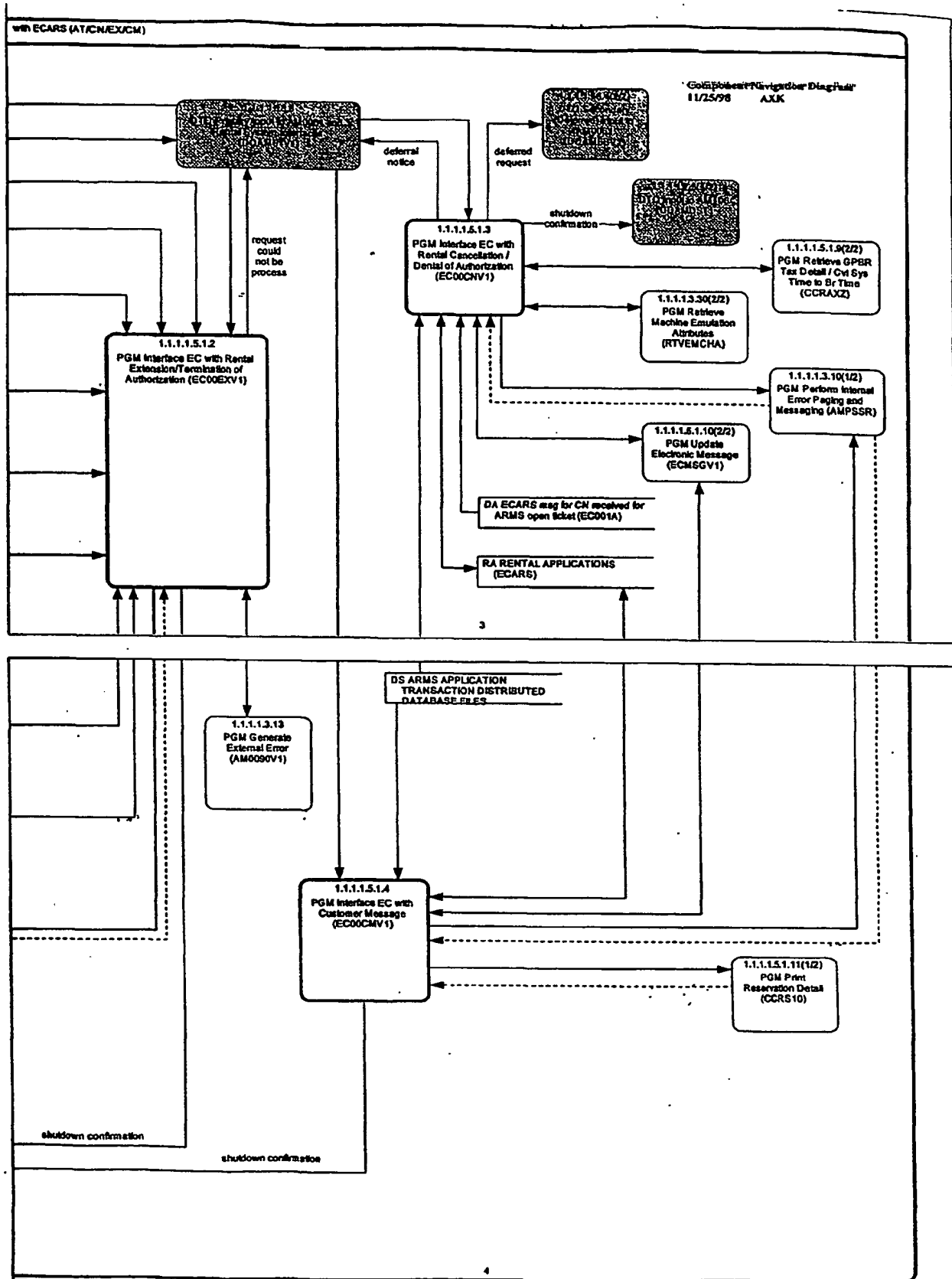
Activity Modes Diagram  
07/15/98

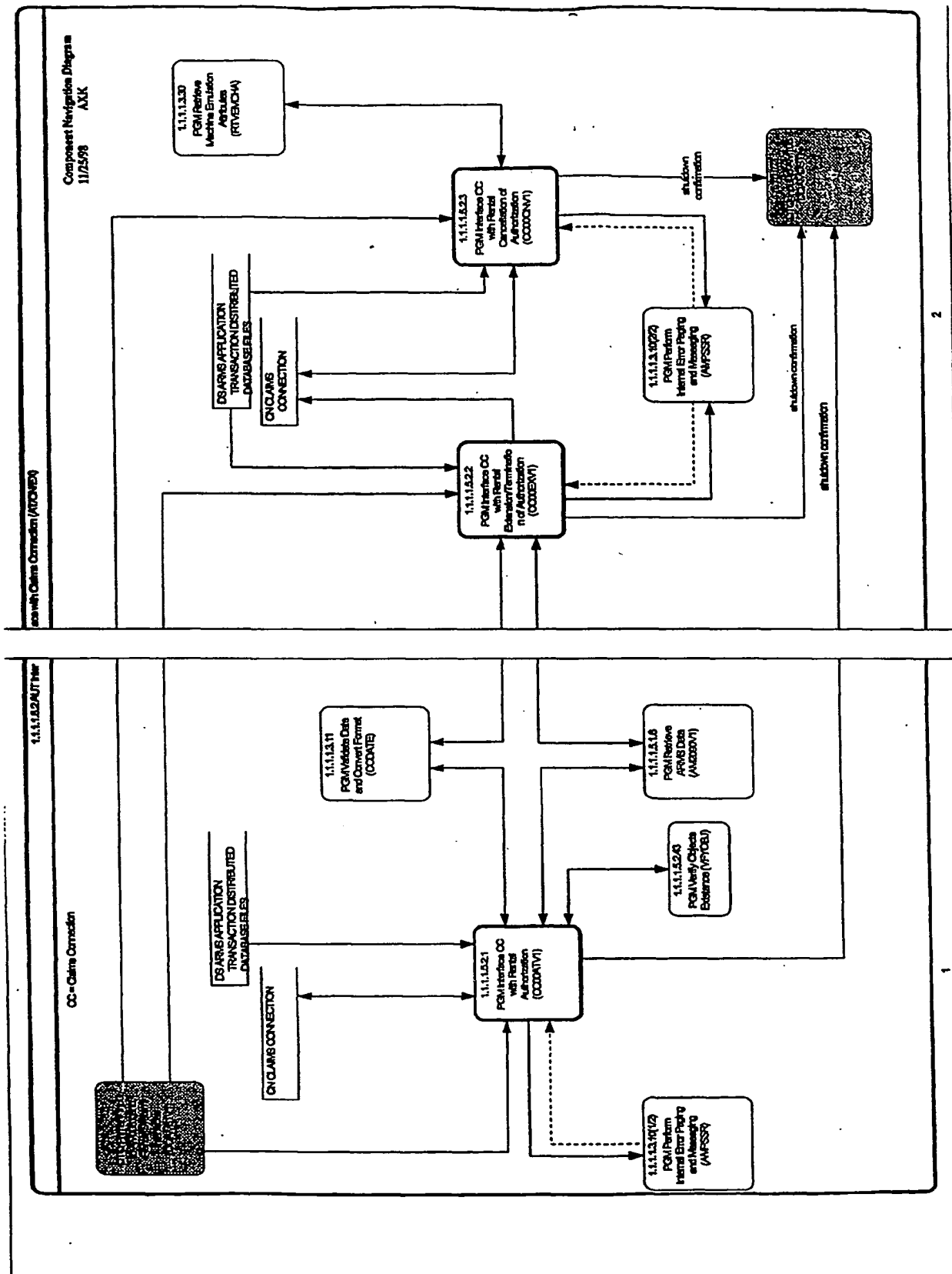
1.1.1.1.5 AUT Interface with Rental Systems (AT/CN/EX/CM)

1.1.1.1.5.1.\*  
AUT Interface with  
ECAPS  
(AT/CN/EX/CM)

1.1.1.1.5.2.\*  
AUT Interface with  
Claims Connection  
(AT/CN/EX)







1.1.1.1.8 AUT Generate Vendor Authorization Maintenance Request (ACRA/RC/RE/RN/TR/VM)

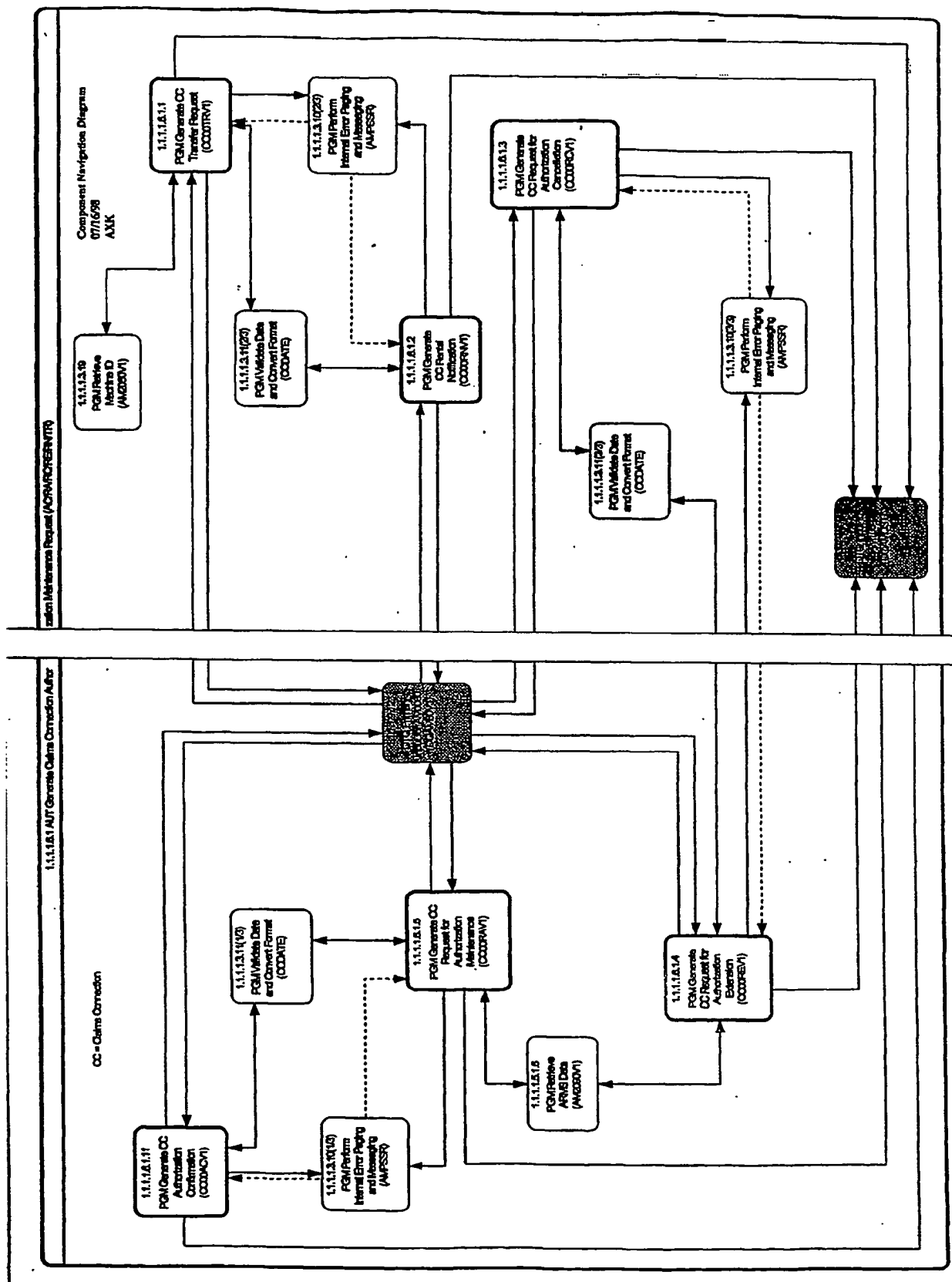
Activity Modes Diagram  
07/16/98  
AXK

1.1.1.1.6.1.\*

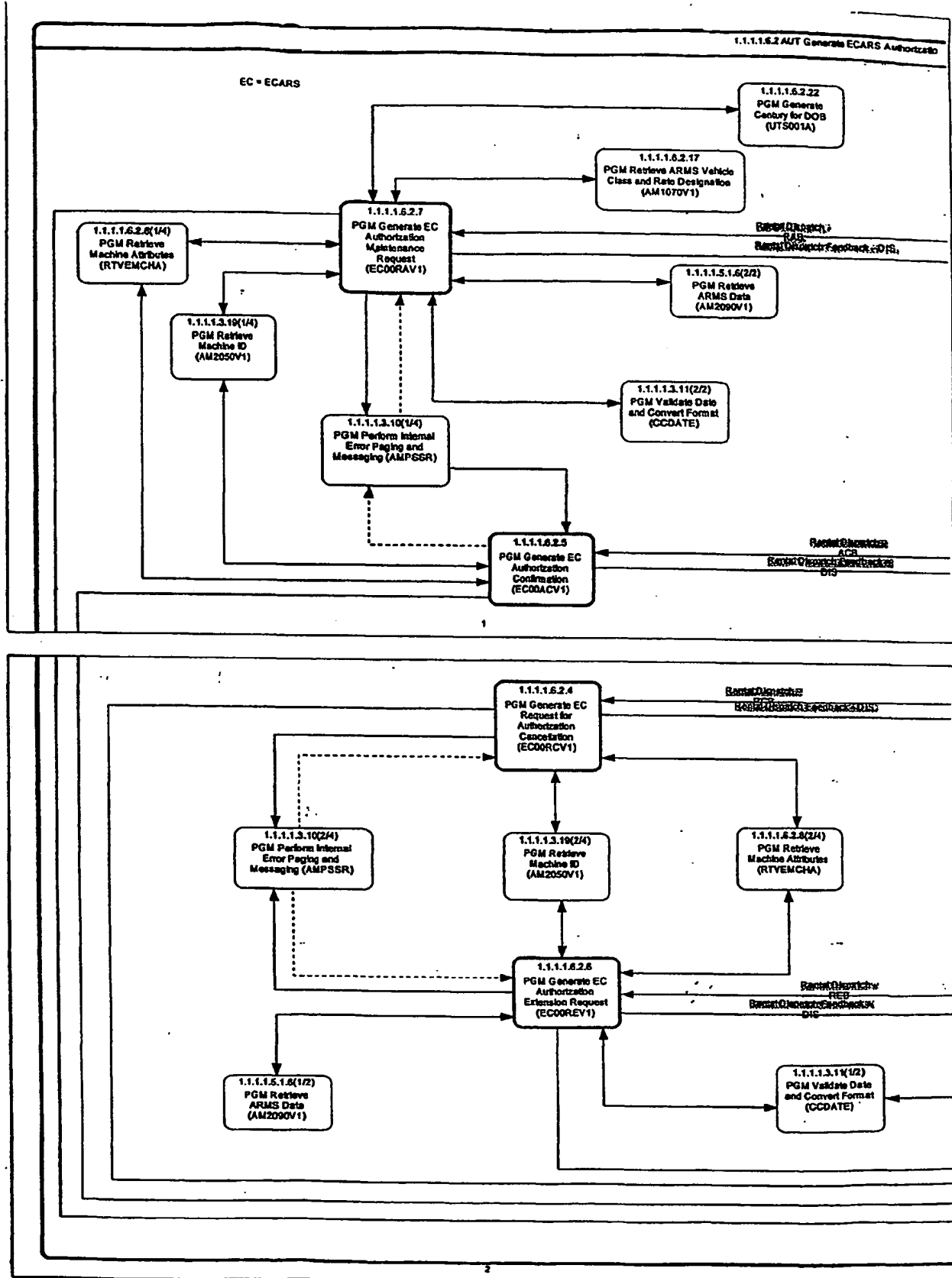
AUT Generate Claims  
Connection Authorization  
Maintenance Request  
(ACRA/RC/RE/RN/TR)

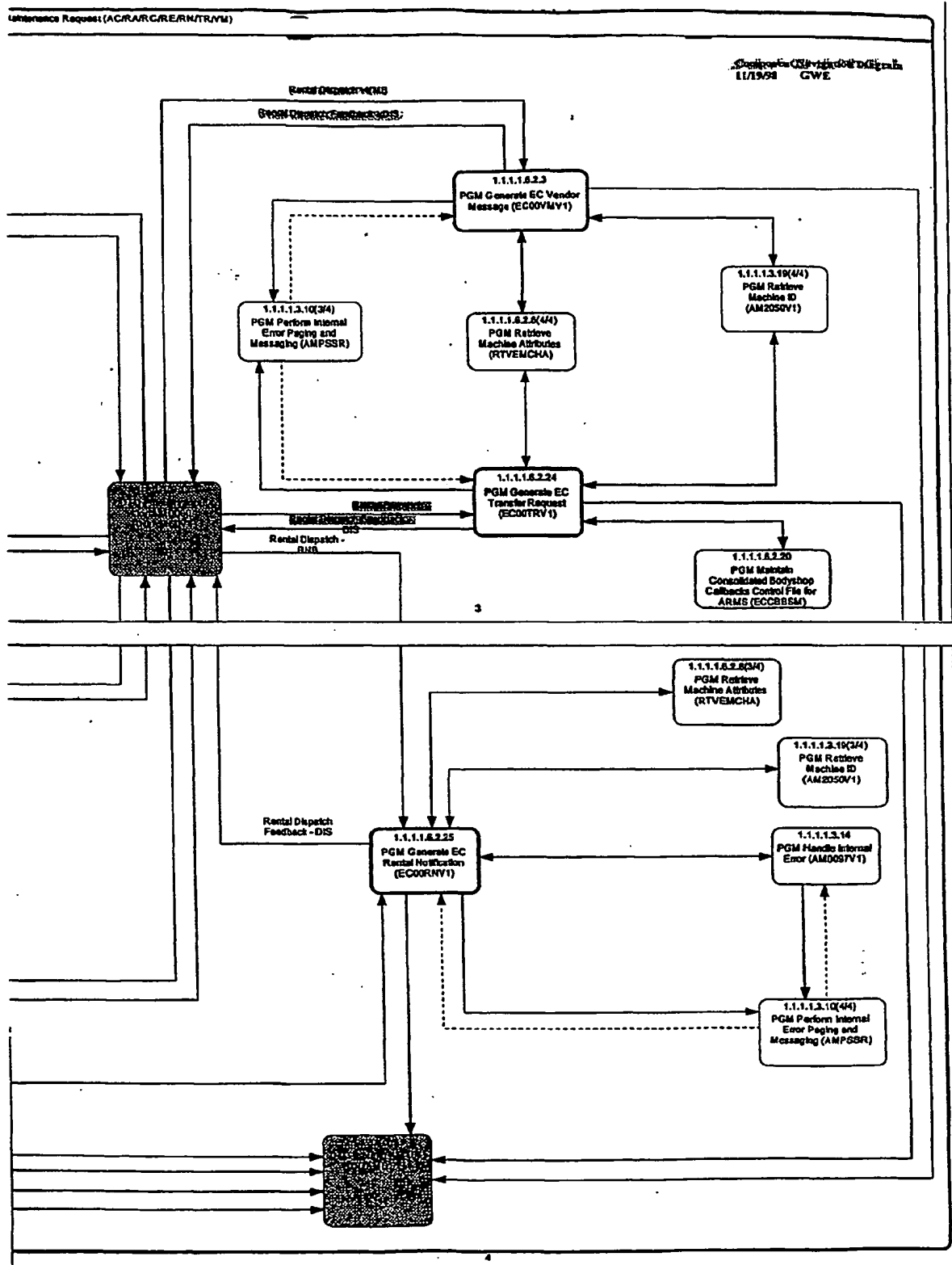
1.1.1.1.6.2.\*

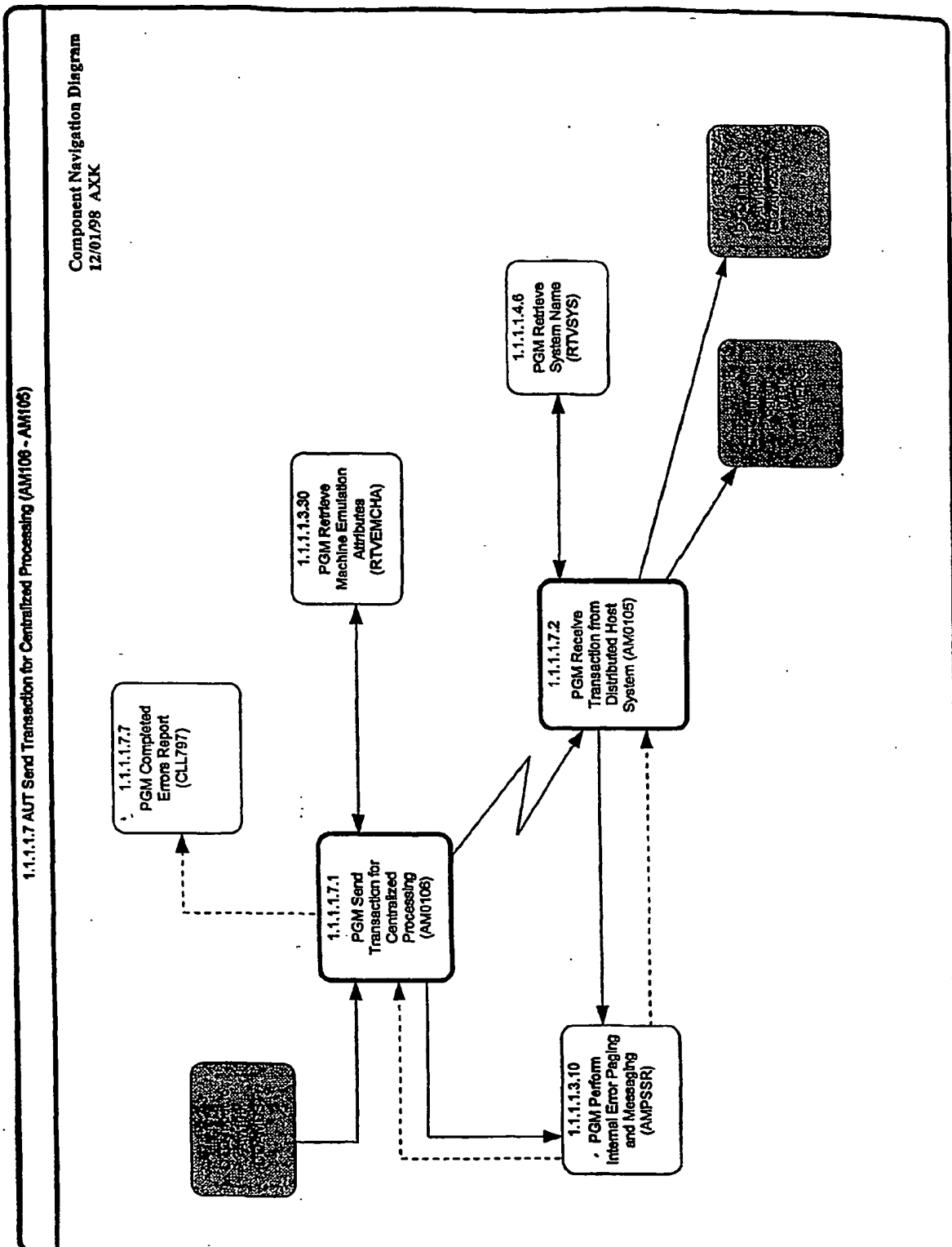
AUT Generate ECARS  
Authorization Maintenance  
Request  
(ACRA/RC/RE/RN/TR/VM)

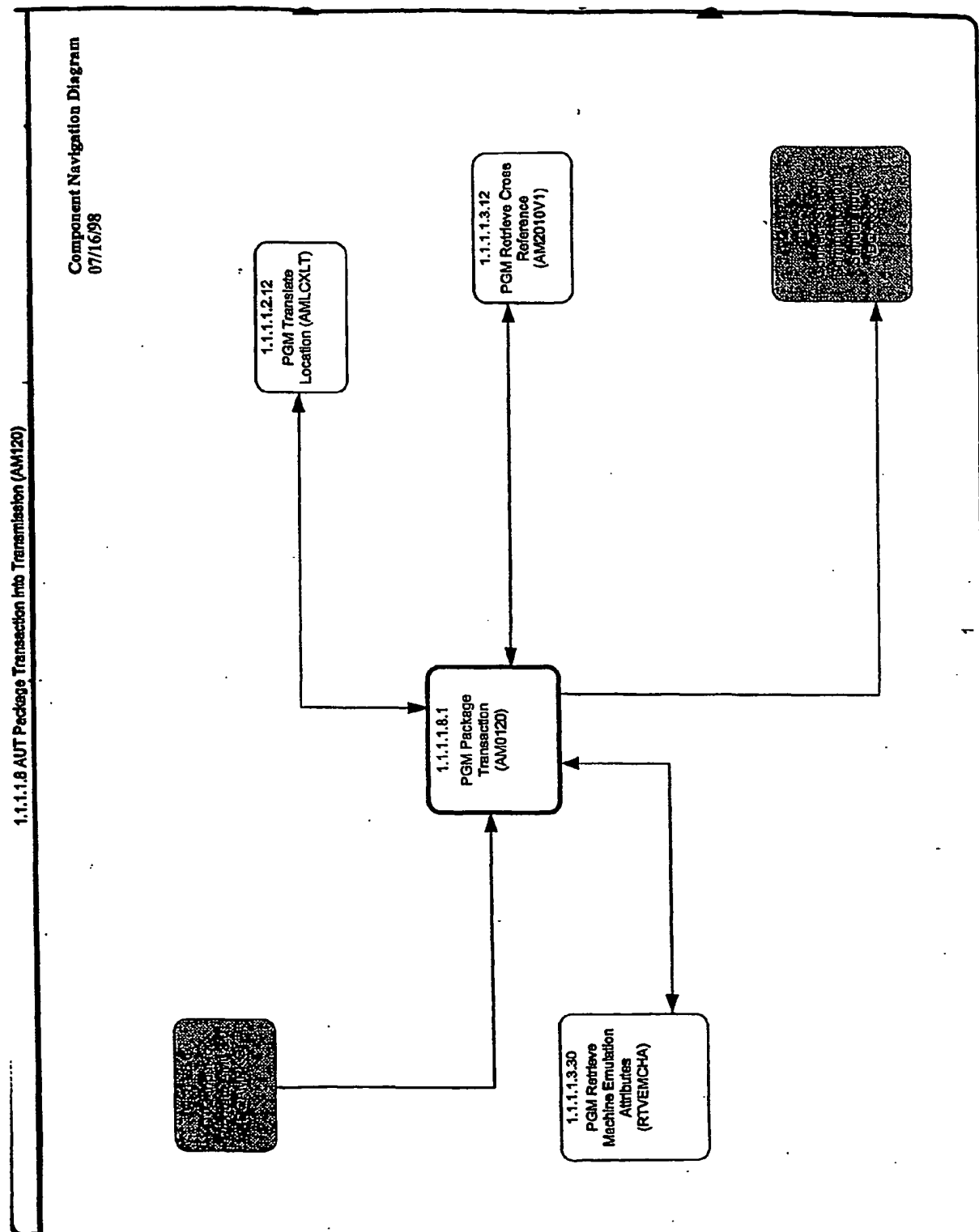


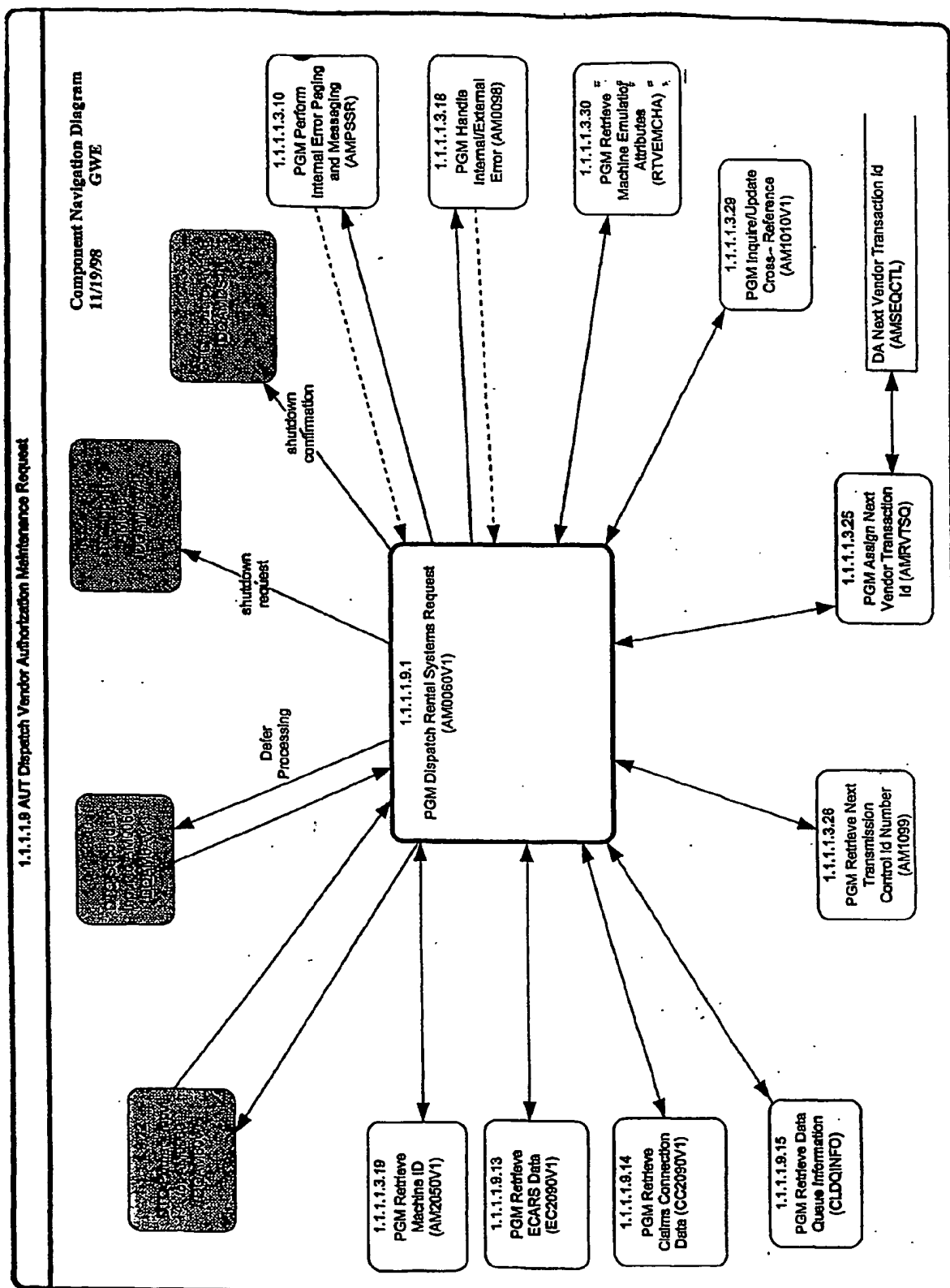




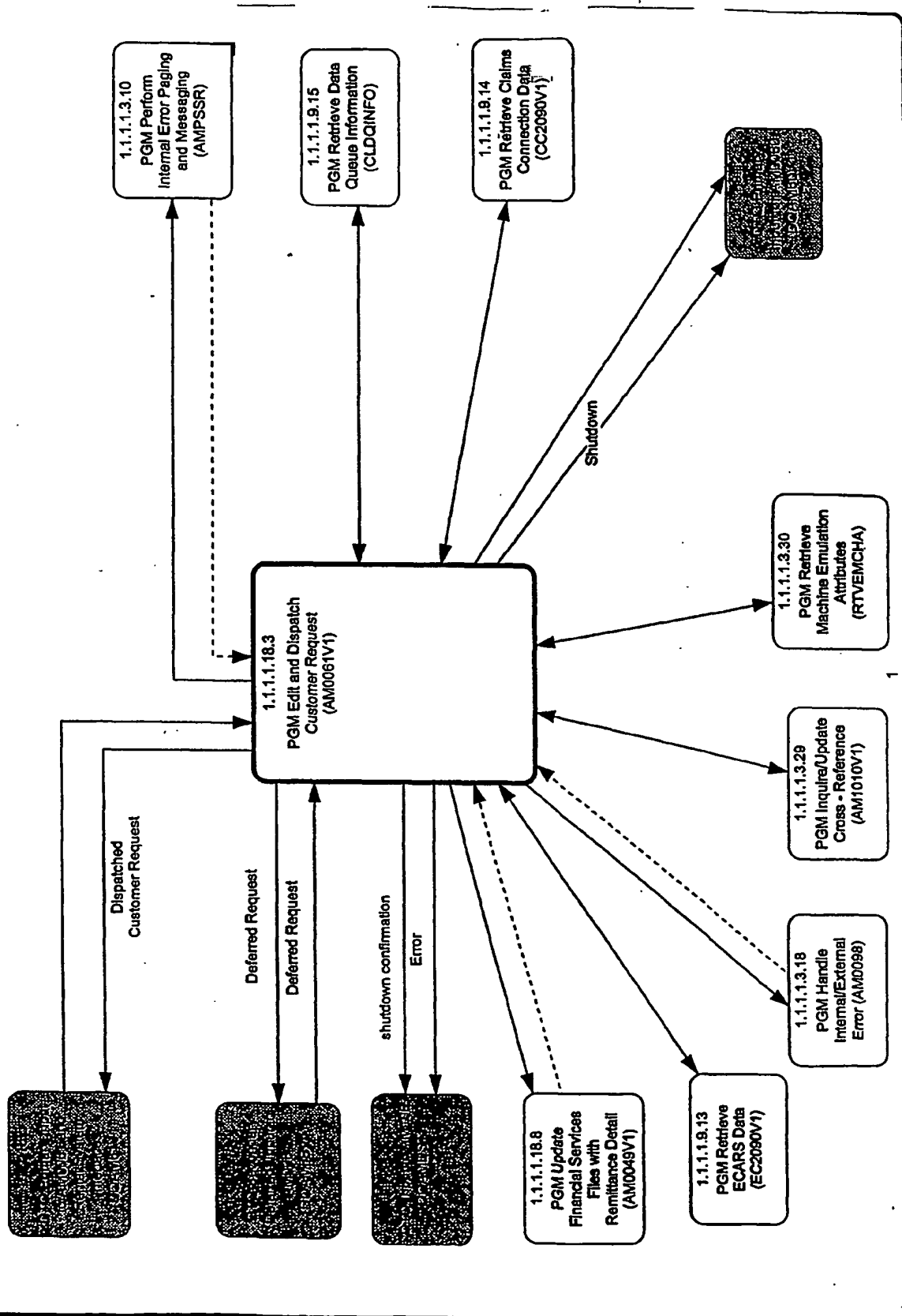


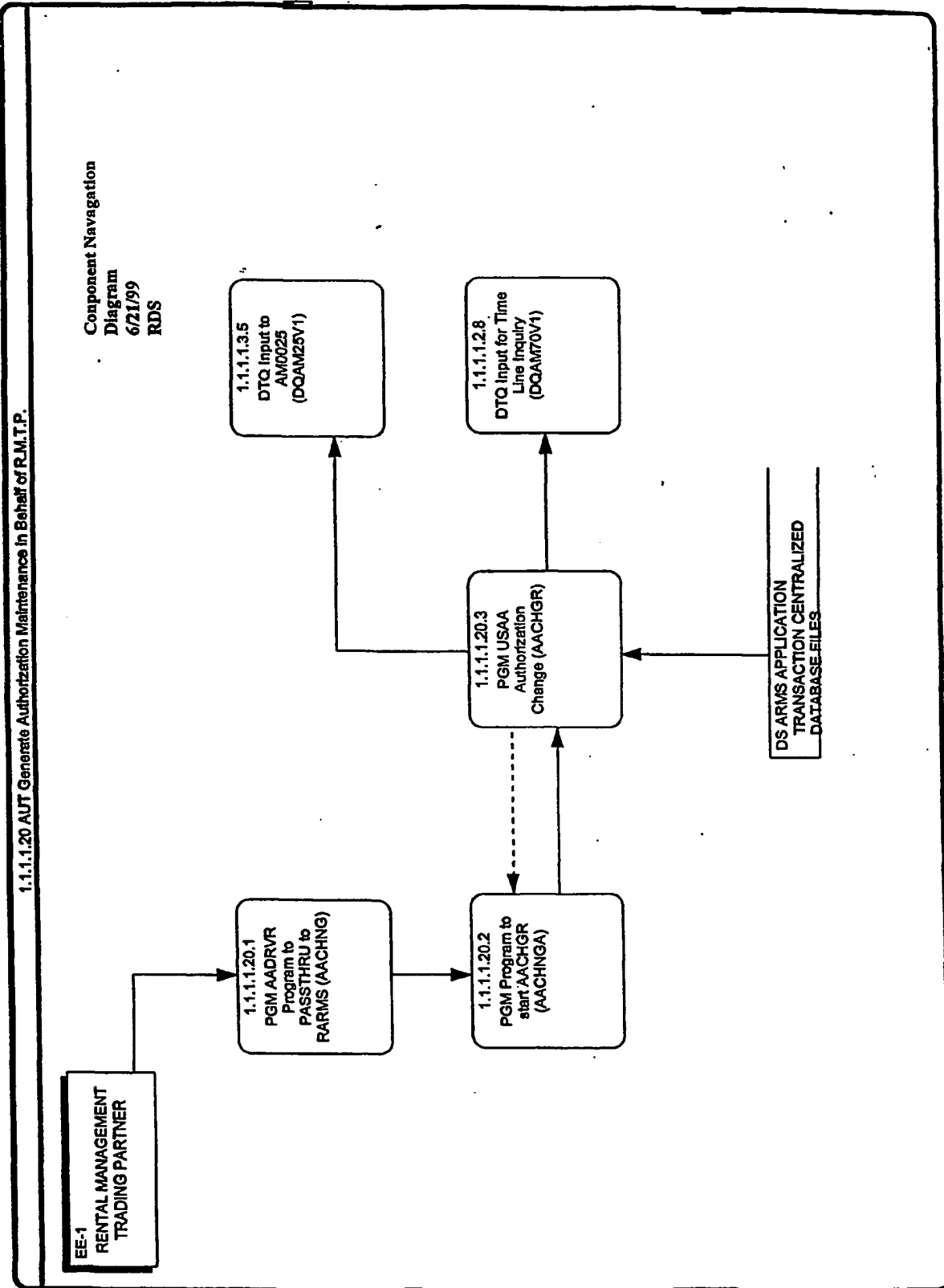


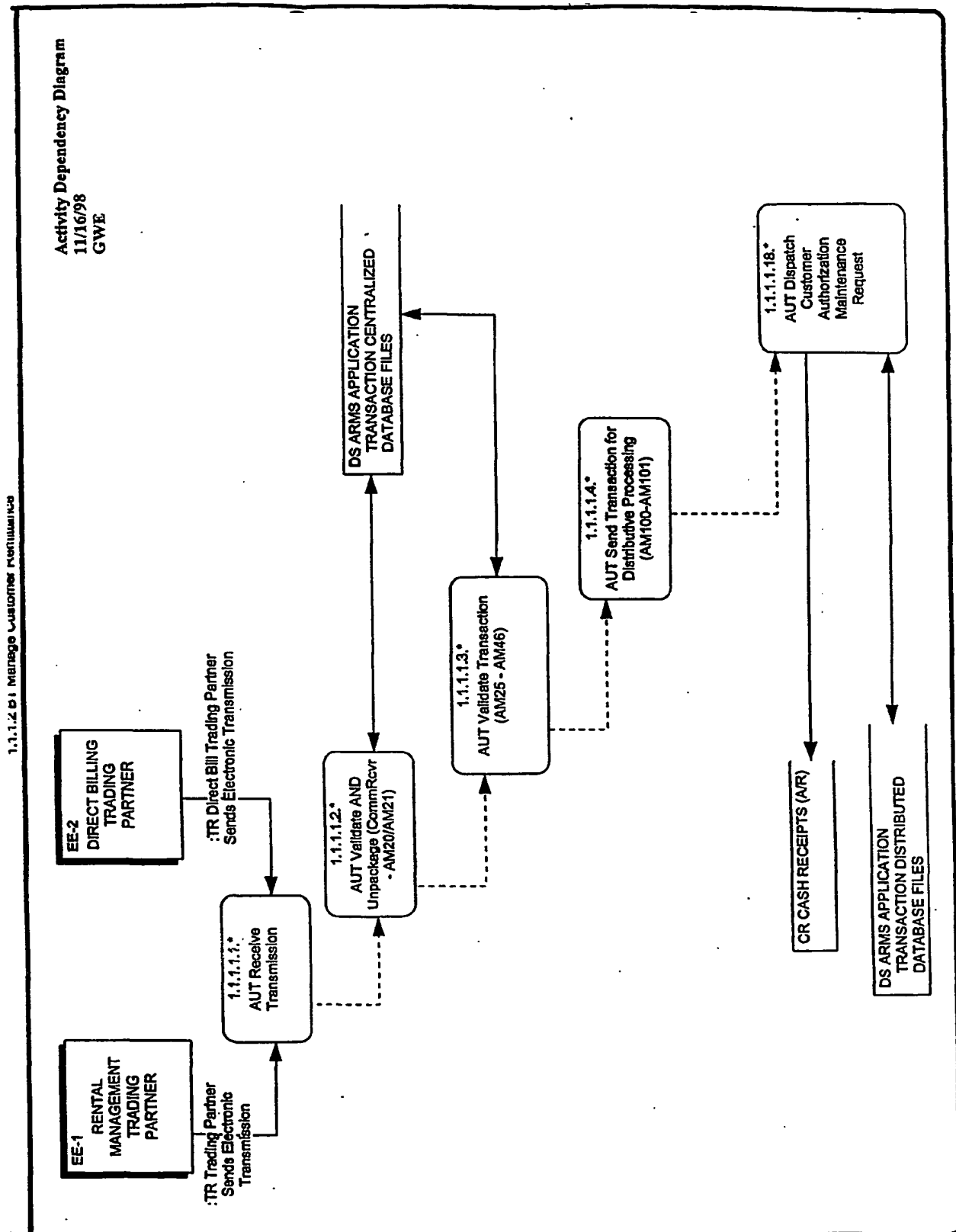




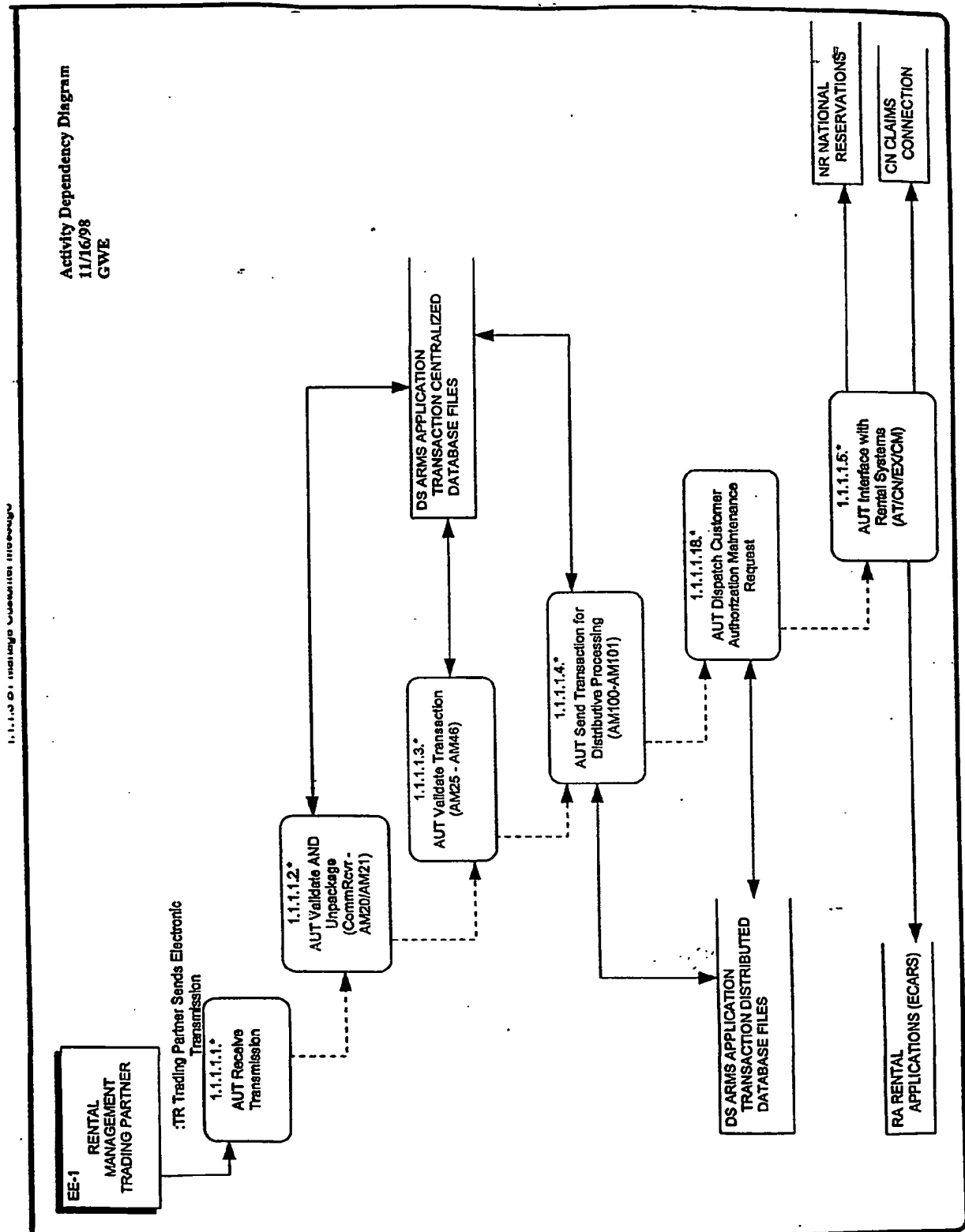
Component Navigation Diagram  
11/30/98 AXK











1.1.2 AA Rental Systems Business Transactions

Business Transactions  
07/16/98

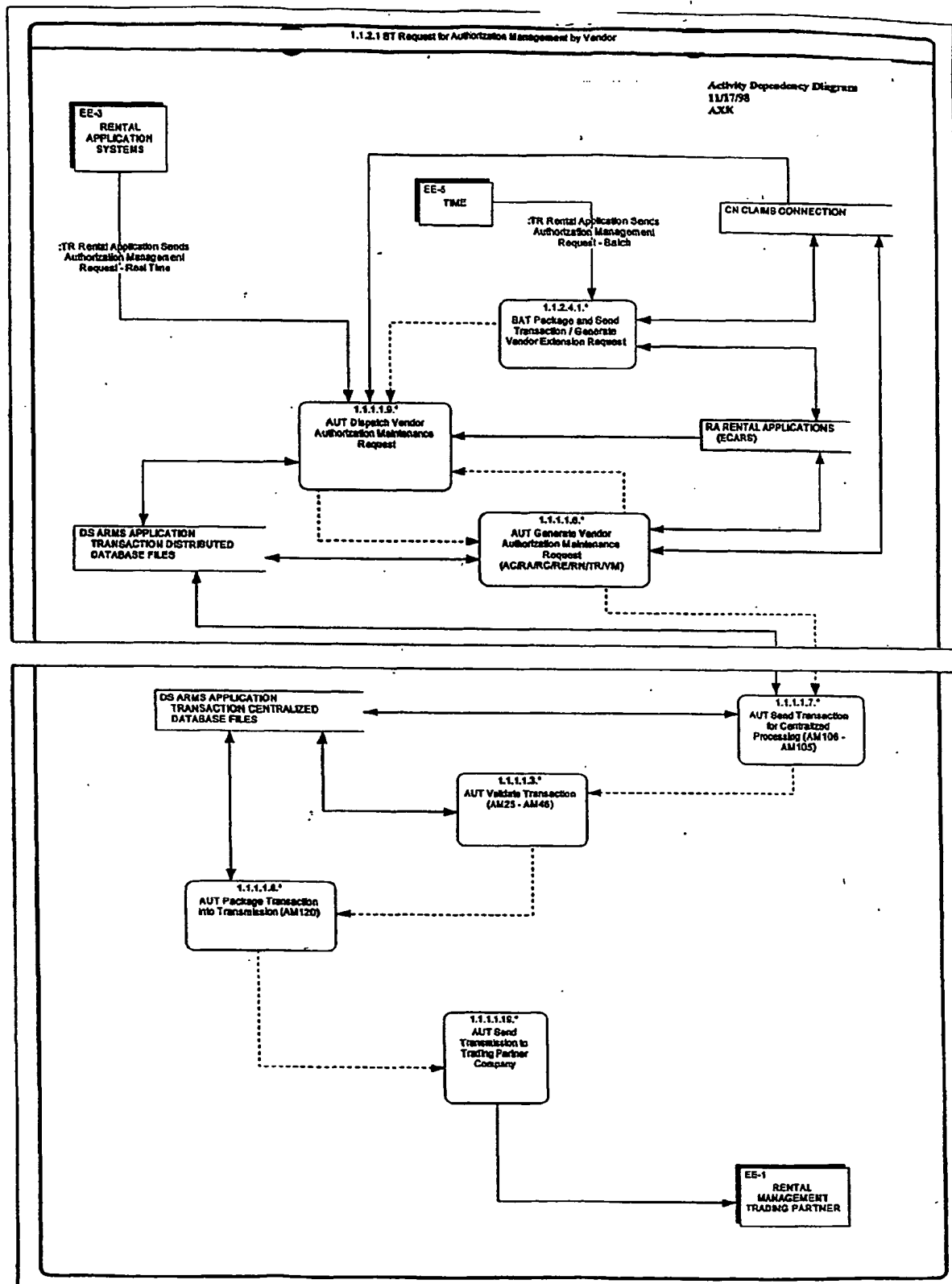
1.1.2.1.\*  
BT Request for  
Authorization  
Management by  
Vendor

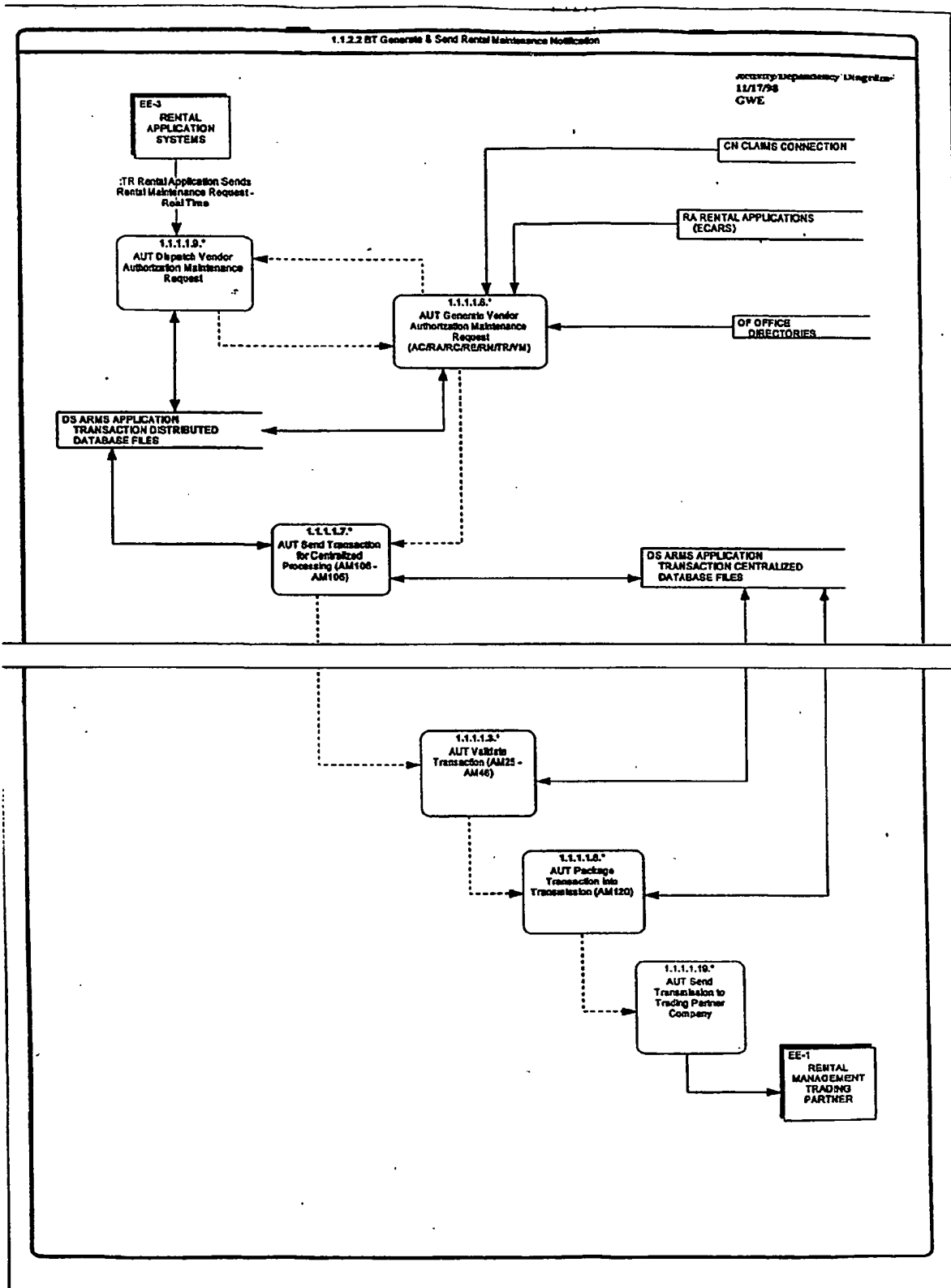
1.1.2.2.\*  
BT Generate & Send  
Rental Maintenance  
Notification

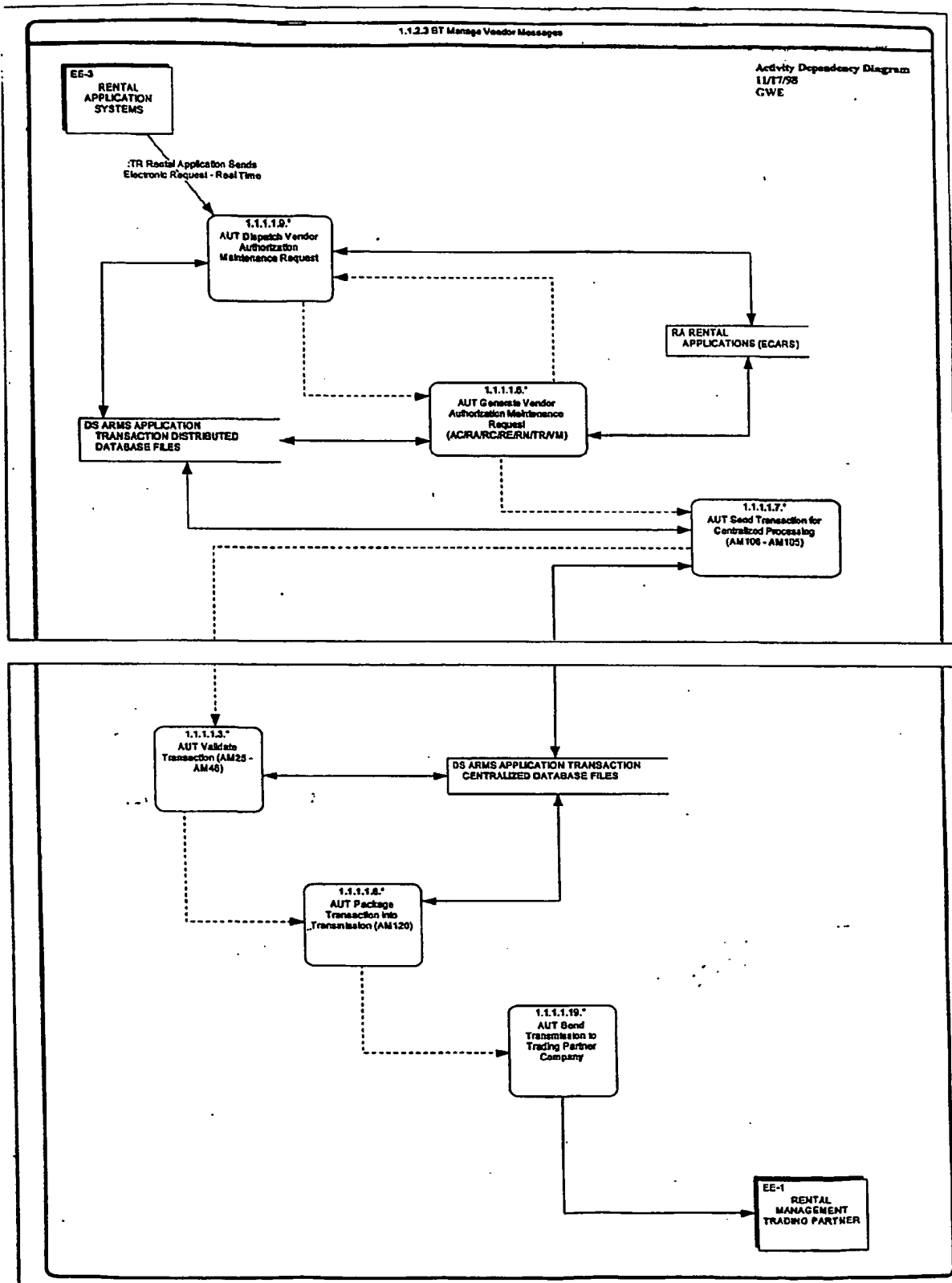
1.1.2.3.\*  
BT Manage Vendor  
Messages

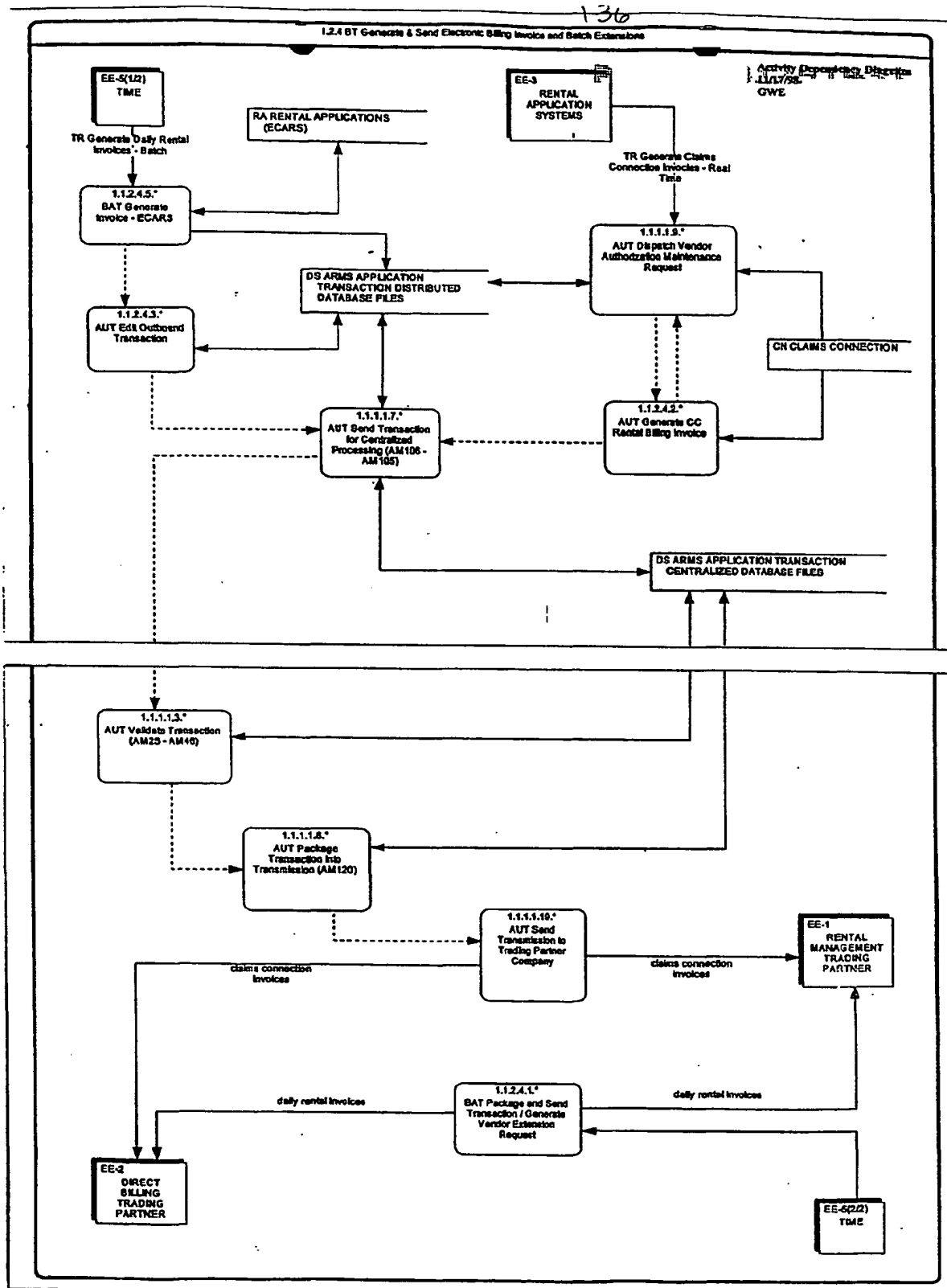
1.1.2.4.\*  
BT Generate &  
Send Electronic  
Billing Invoice and  
Batch Extensions

1.1.2.5.\*  
BT Close ARMS  
Authorized Ticket









1.1.2.4.1 BAT Package and Send Transaction / Generate Vendor Extension Request

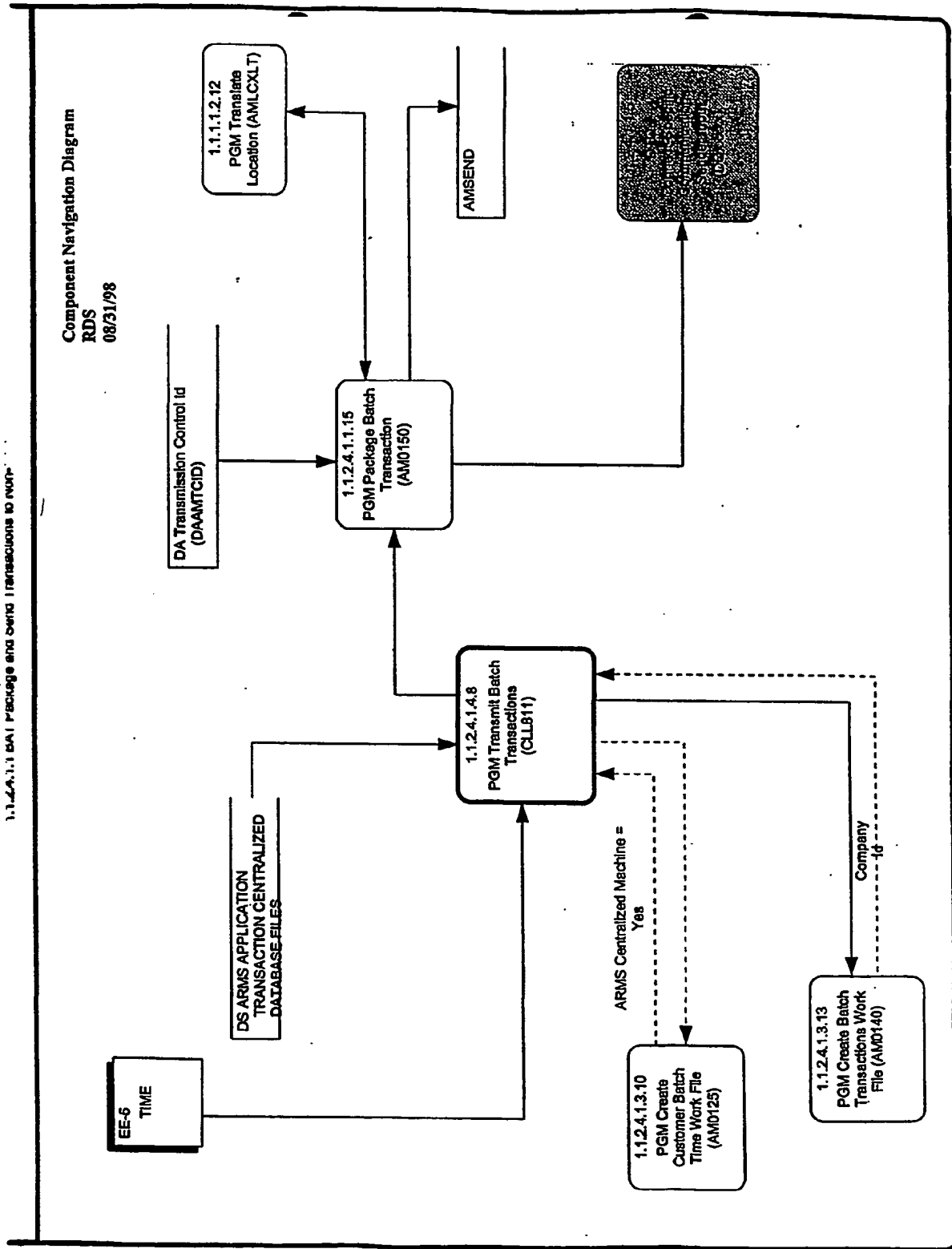
Activity Modes Diagram  
AXK  
07/13/98

1.1.2.4.1.1.\*  
BAT Package and  
Send Transactions  
to Non-\*\*\*\*\*

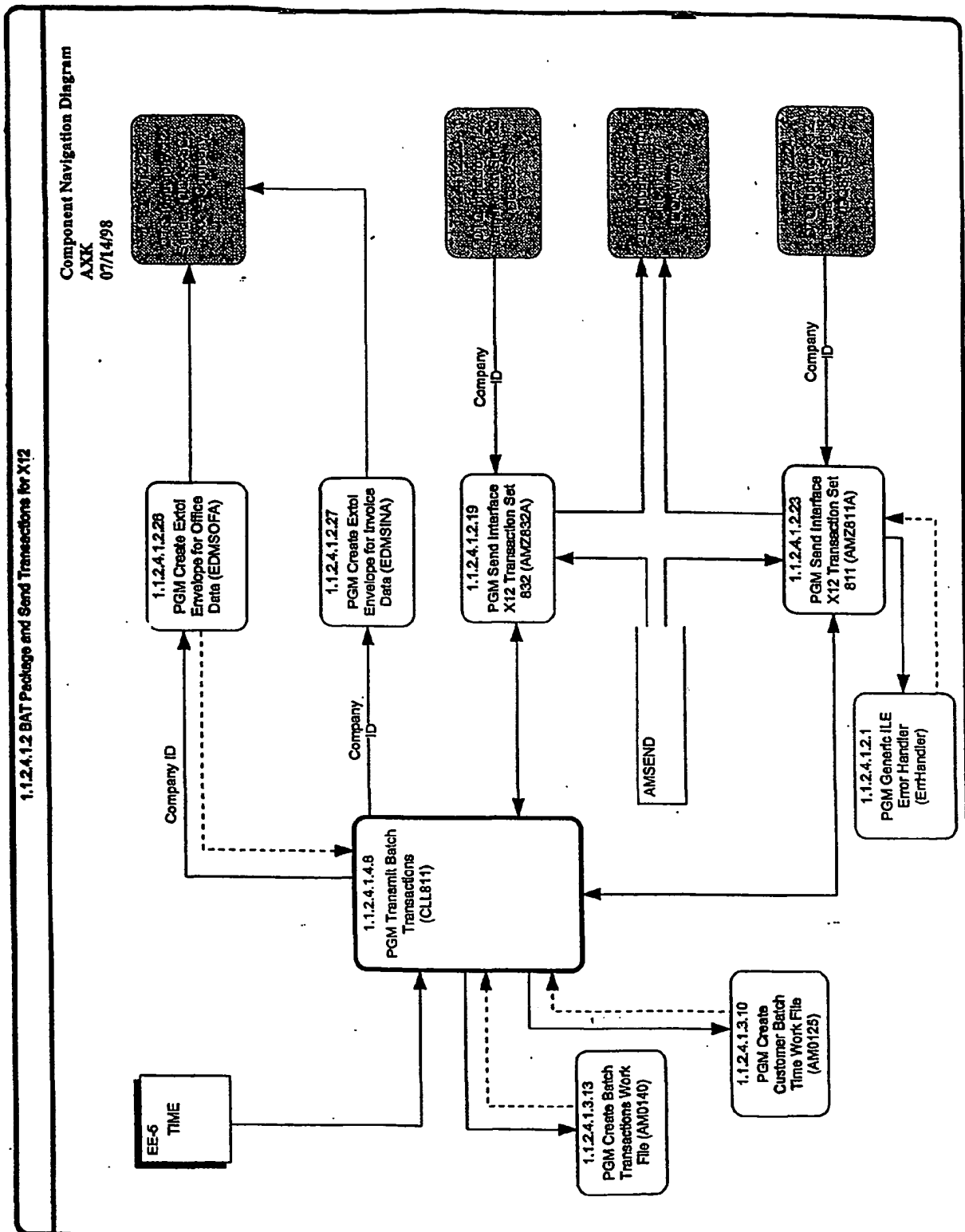
1.1.2.4.1.3.\*  
BAT Package and  
Send Transactions  
to \*\*\*\*\*

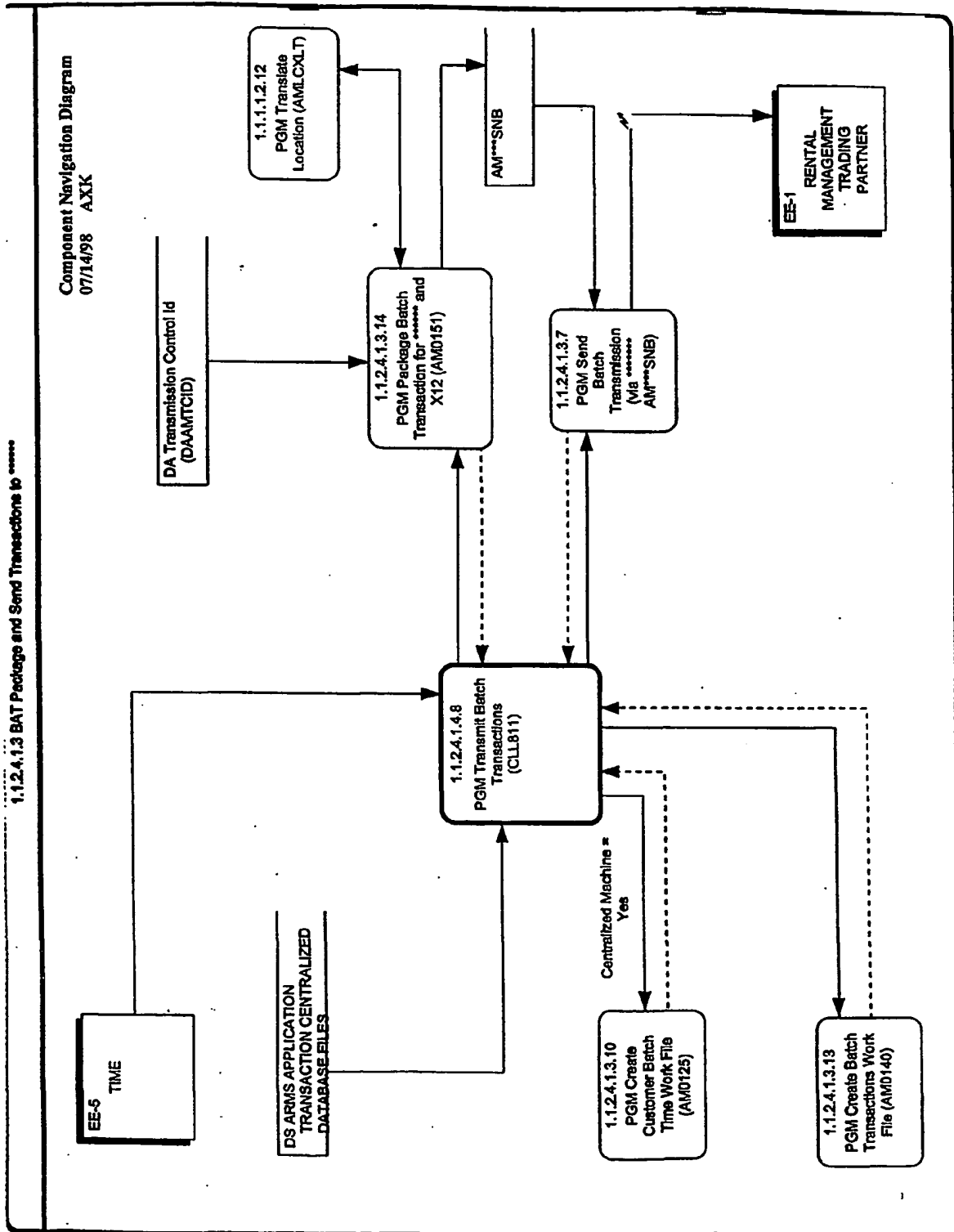
1.1.2.4.1.2.\*  
BAT Package and  
Send Transactions  
for X12

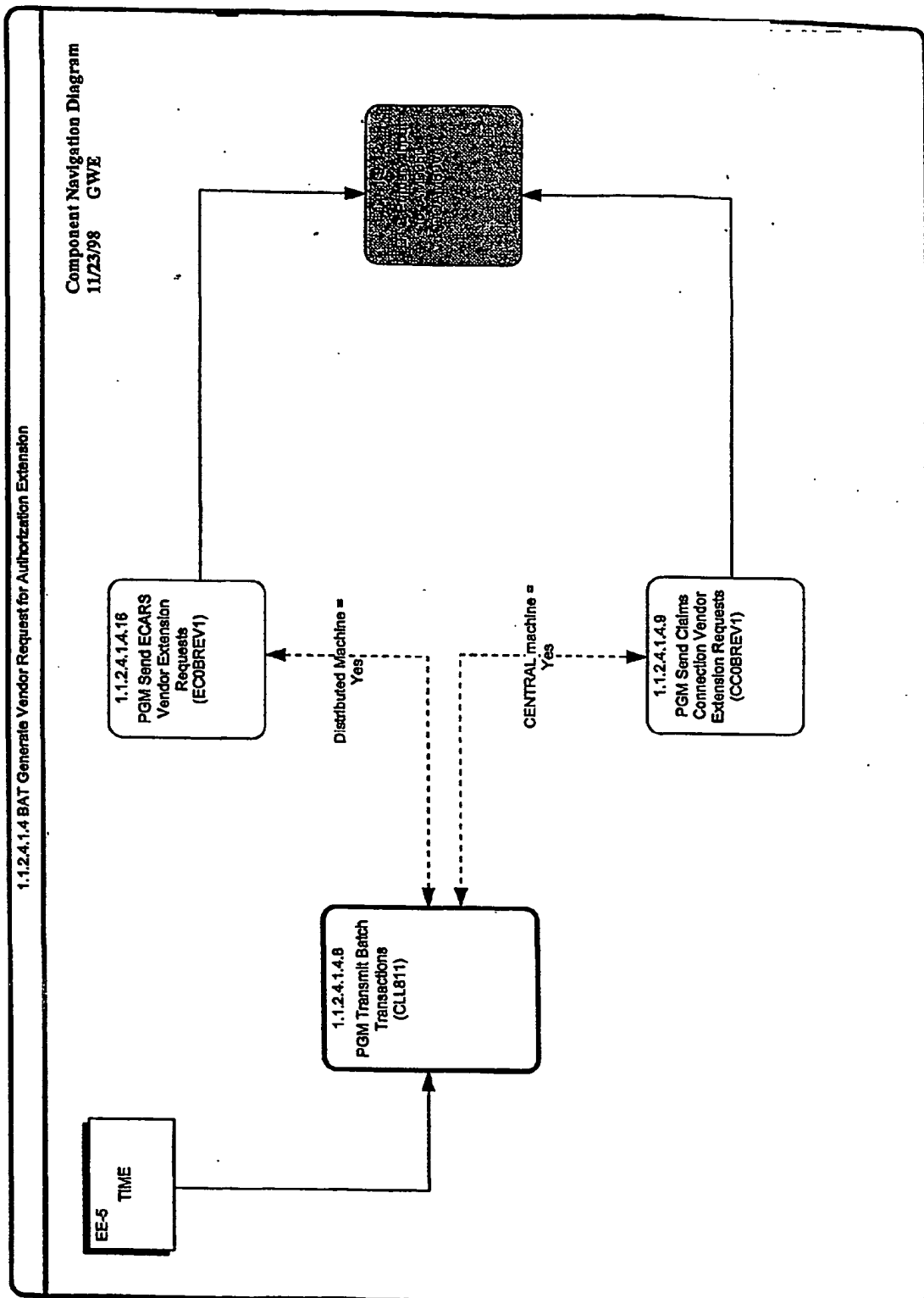
1.1.2.4.1.4.\*  
BAT Generate Vendor  
Request for Authorization  
Extension

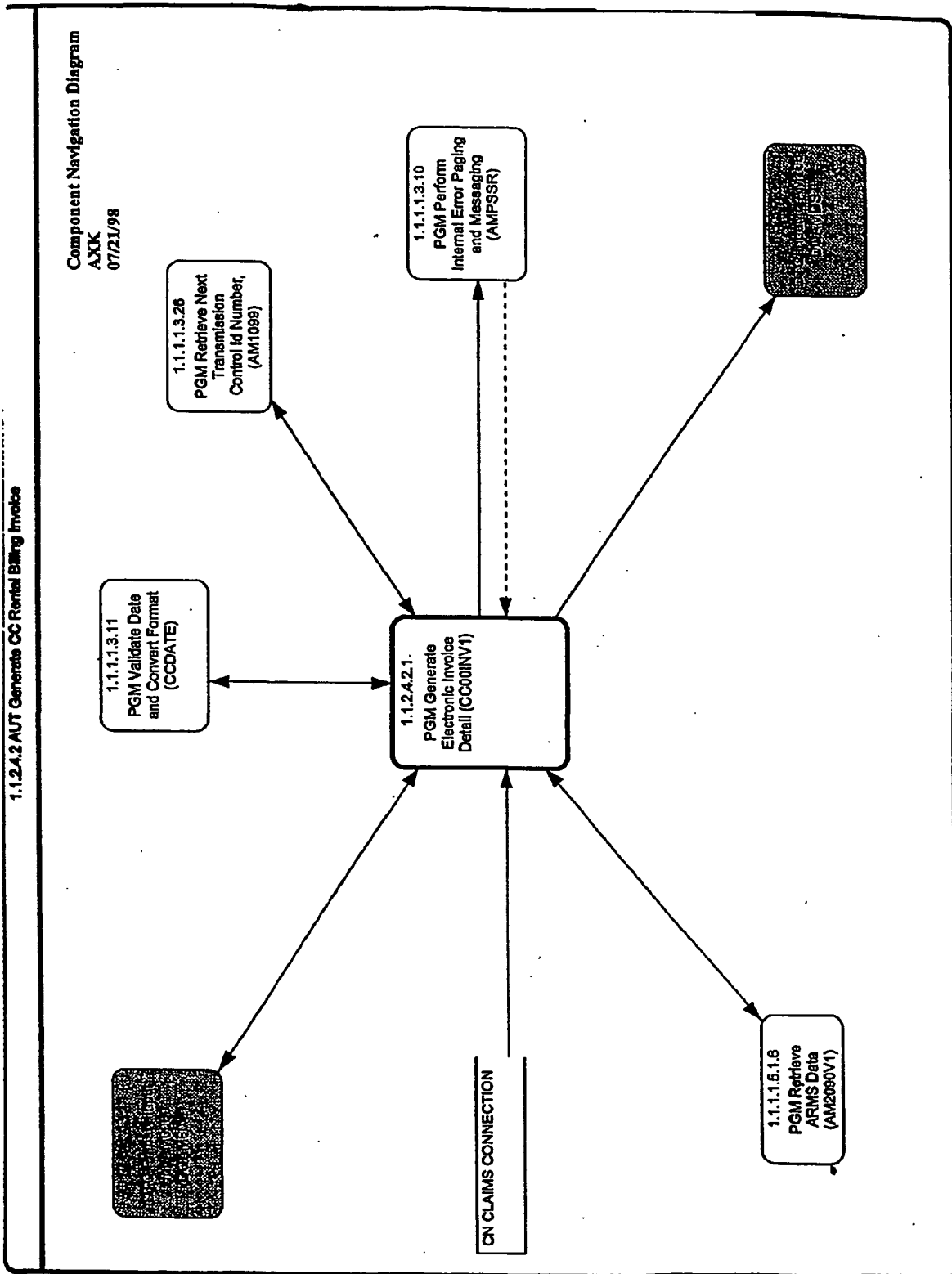


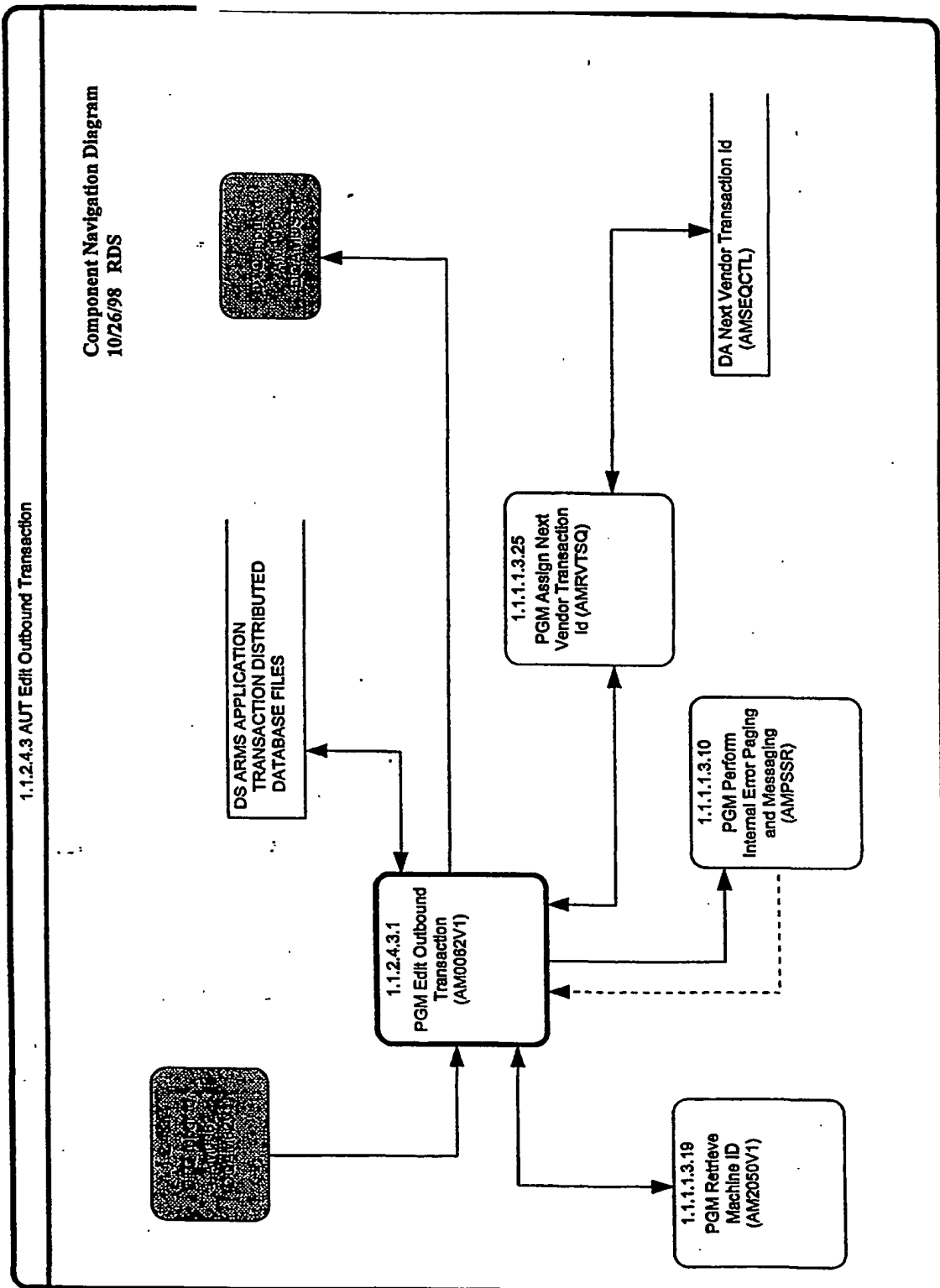


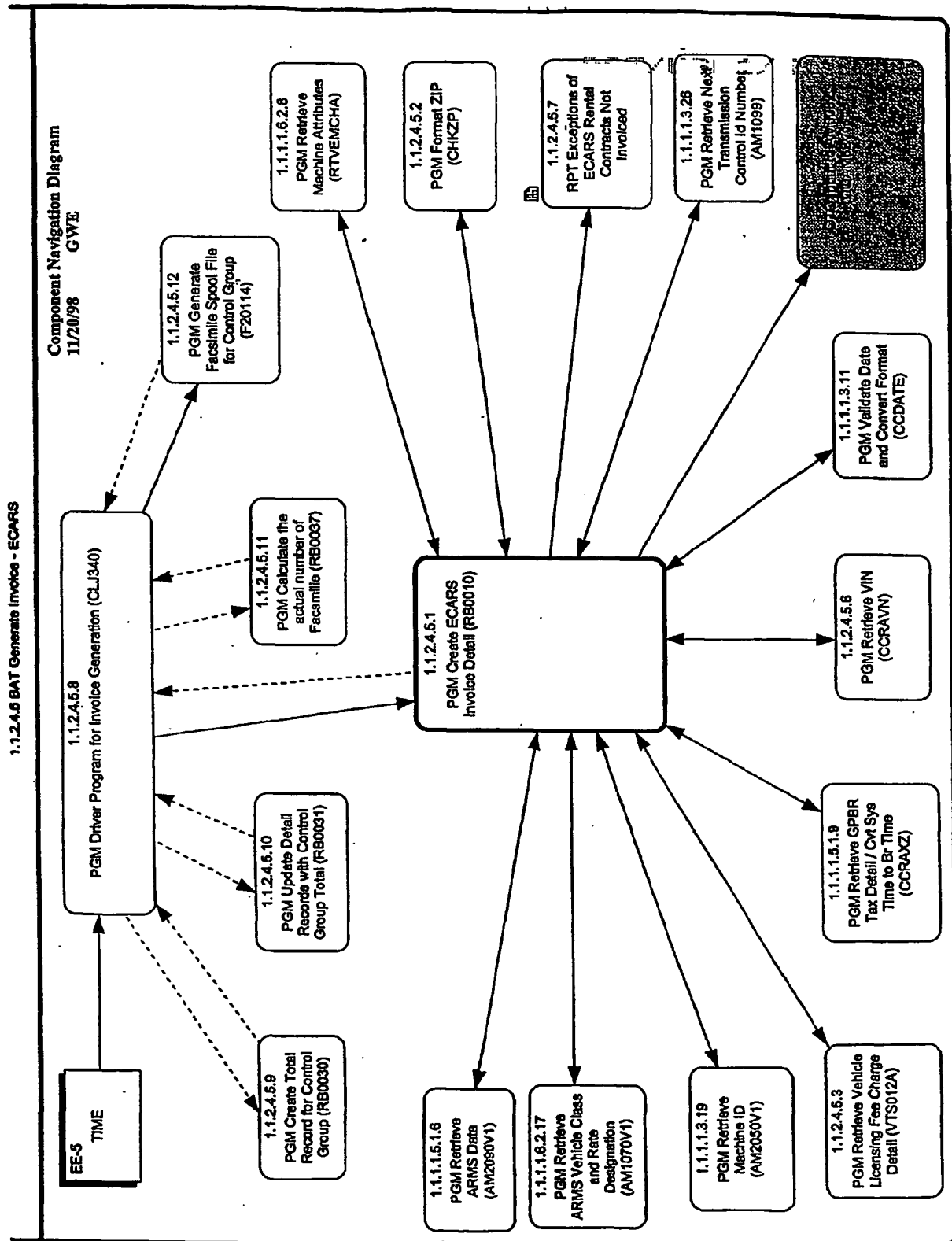


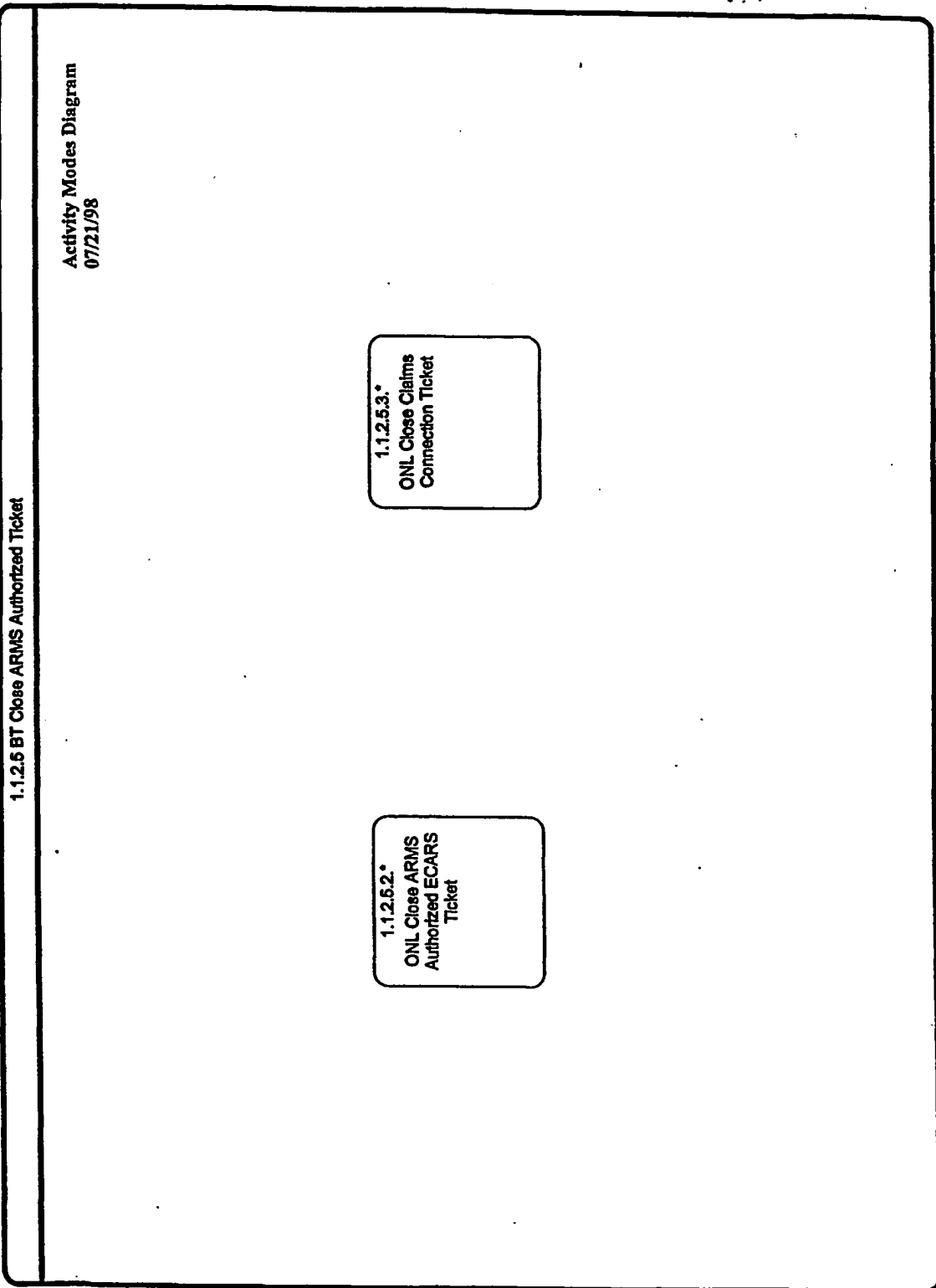


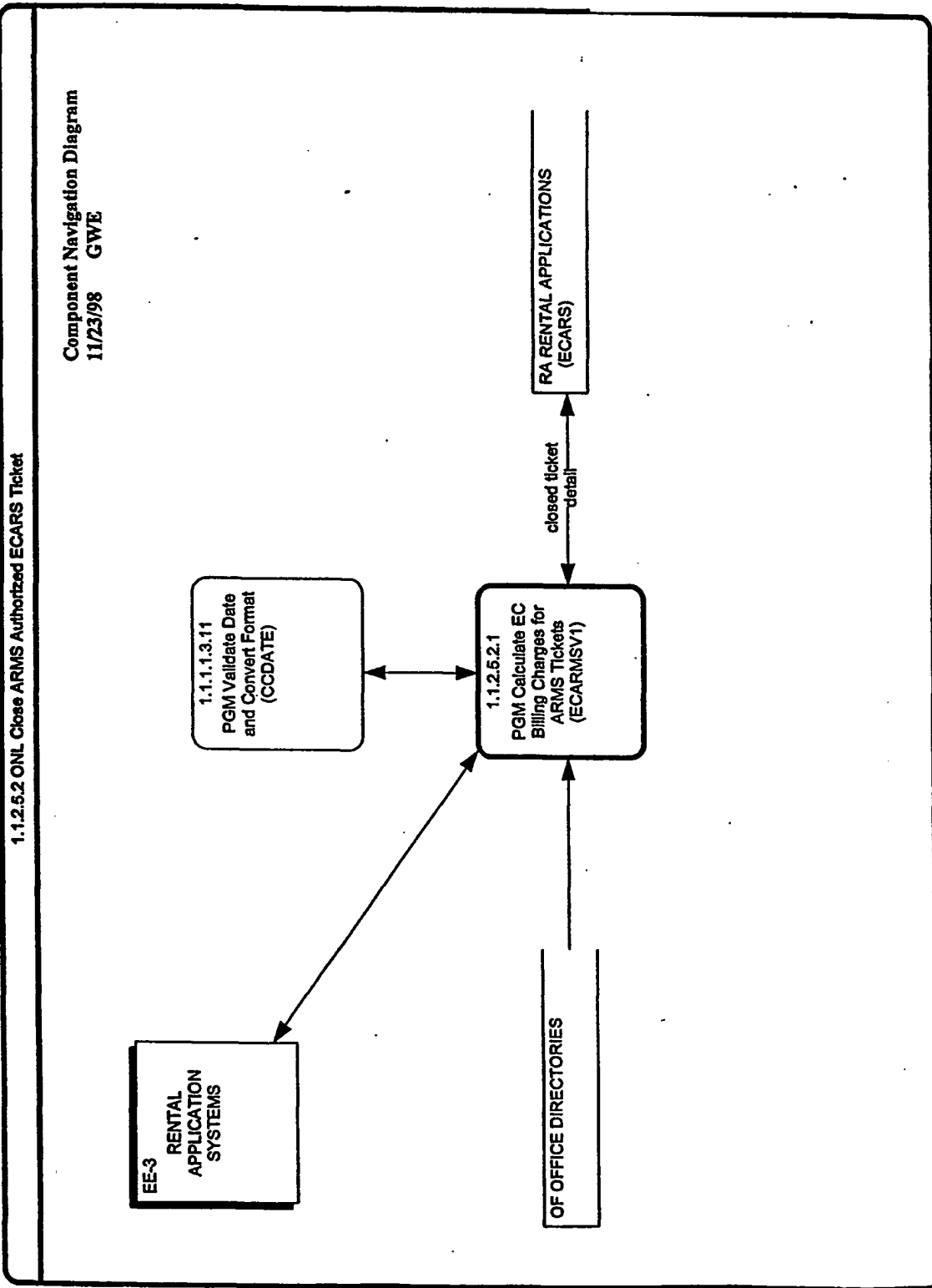




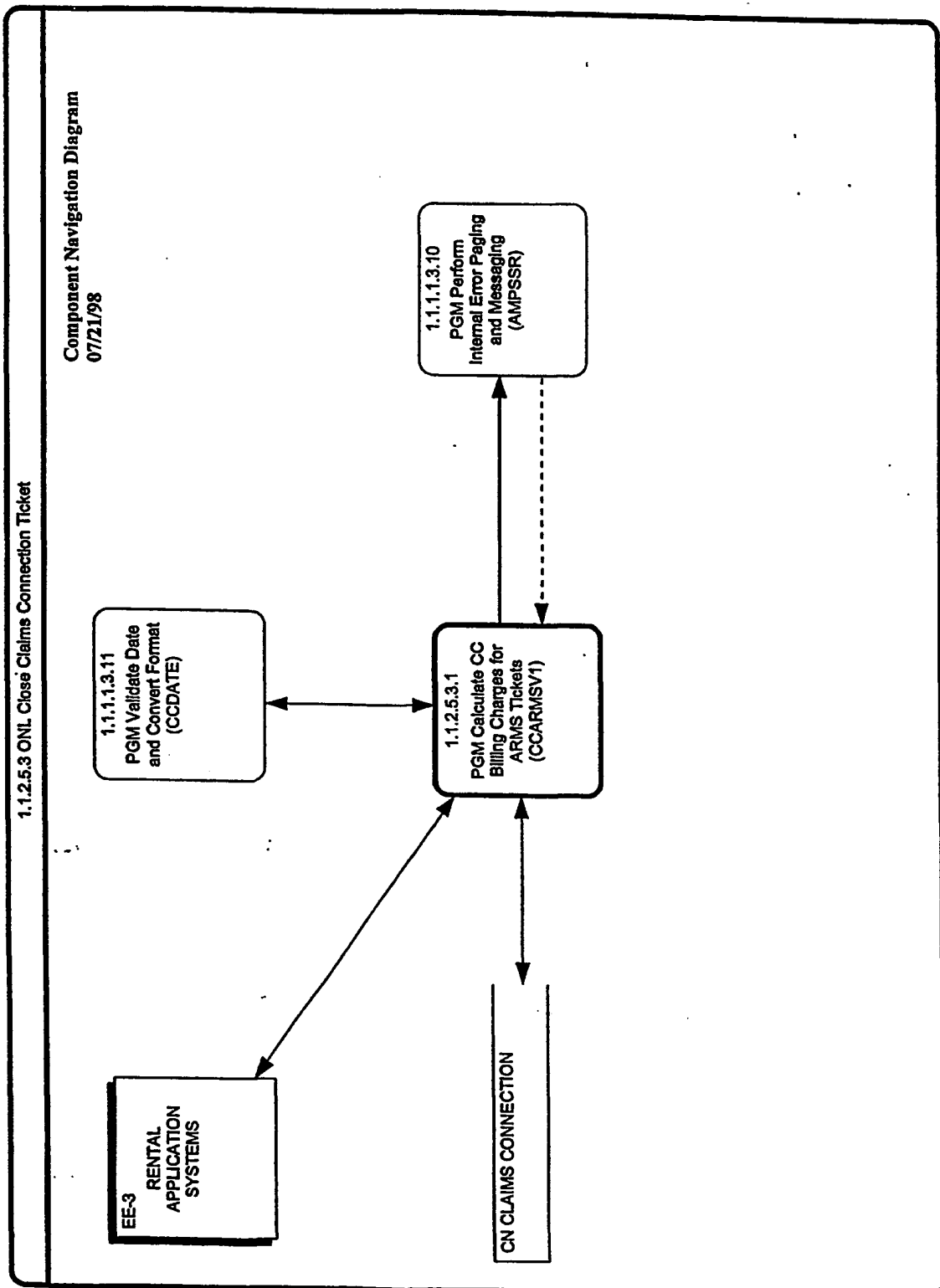












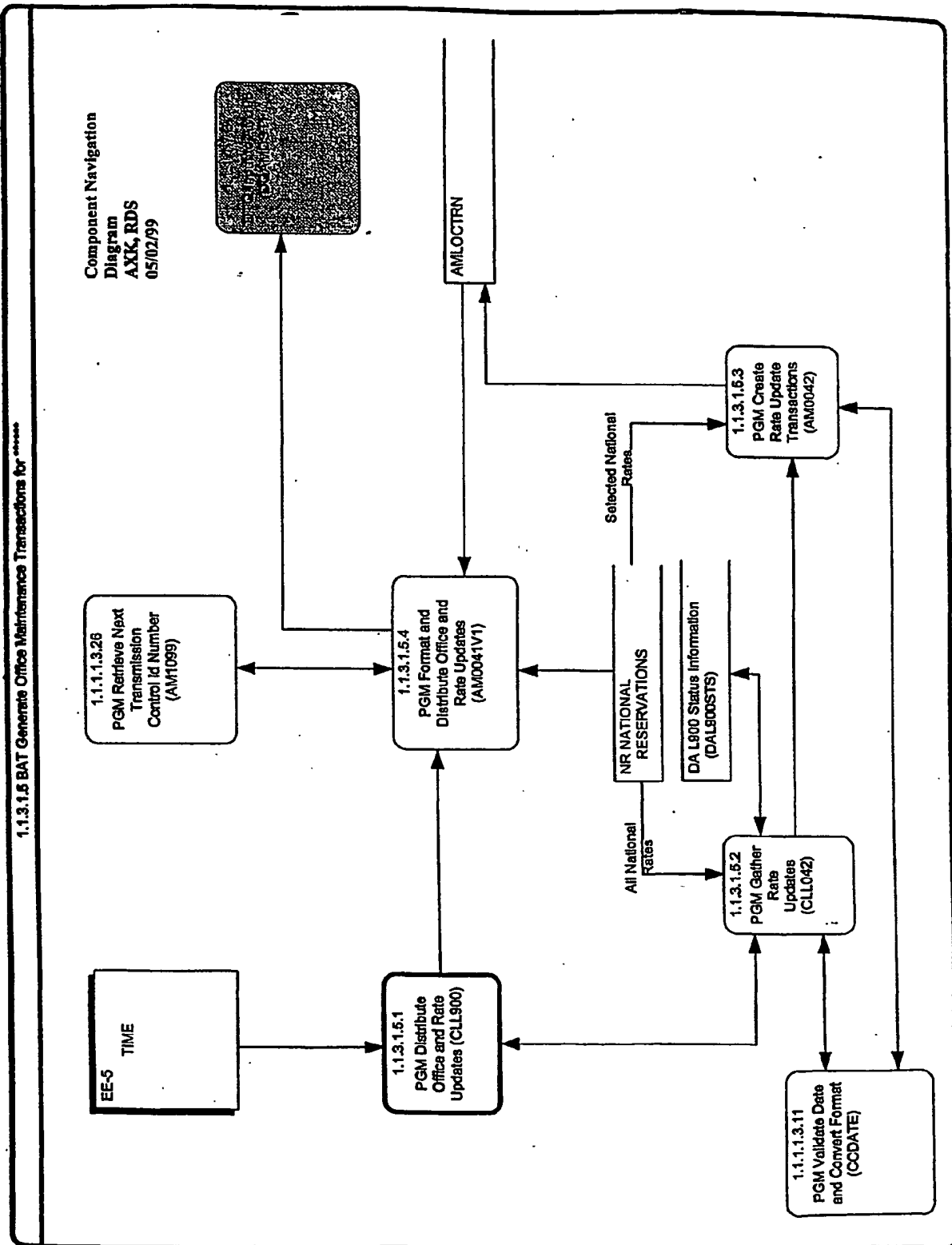
1.1.3 AA Office Information Synchronization between Enterprise and Trading Partner

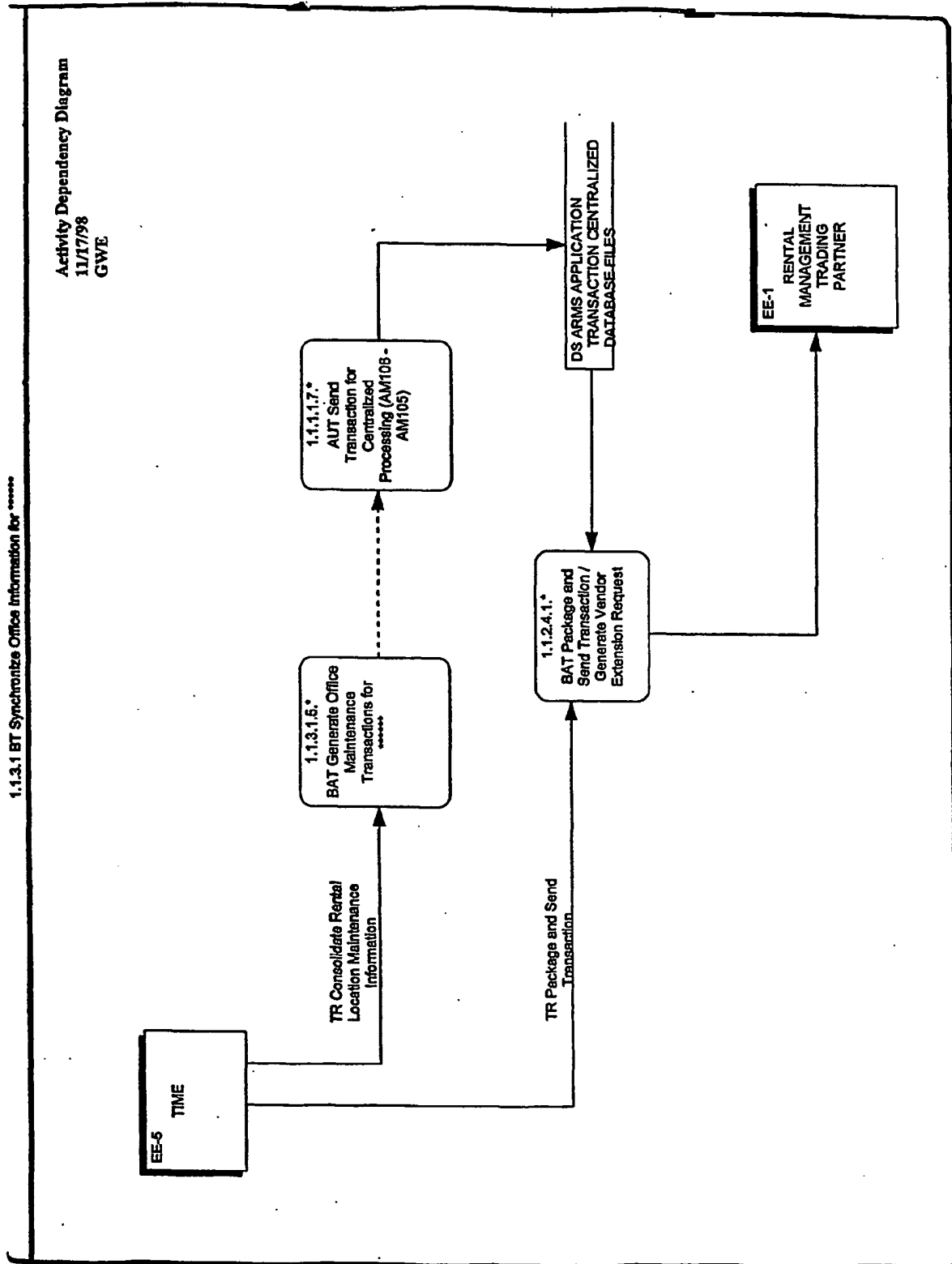
Business Transactions  
AXIK  
07/13/98

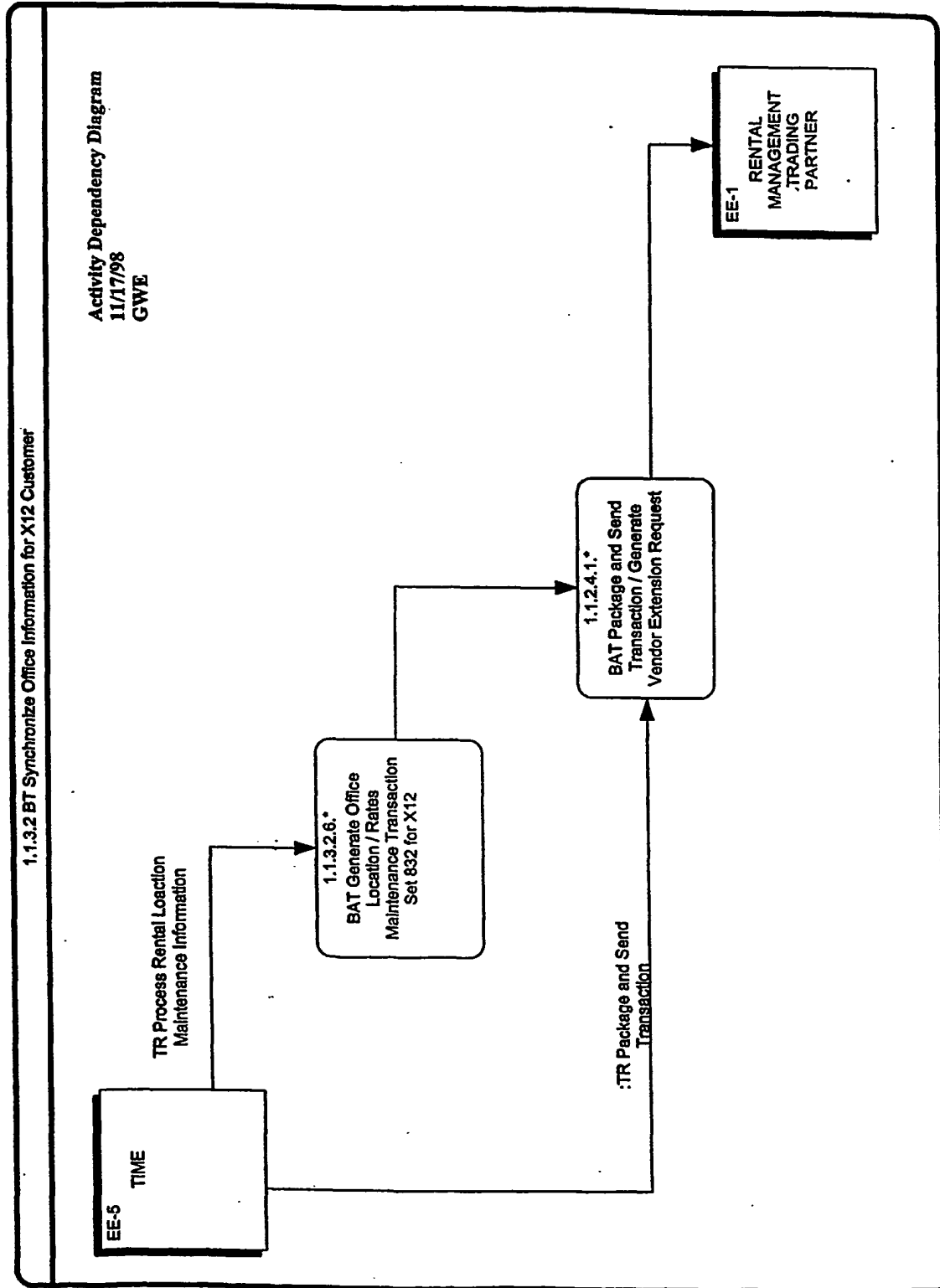
1.1.3.1.\*  
BT Synchronize  
Office Information for  
\*\*\*\*\*

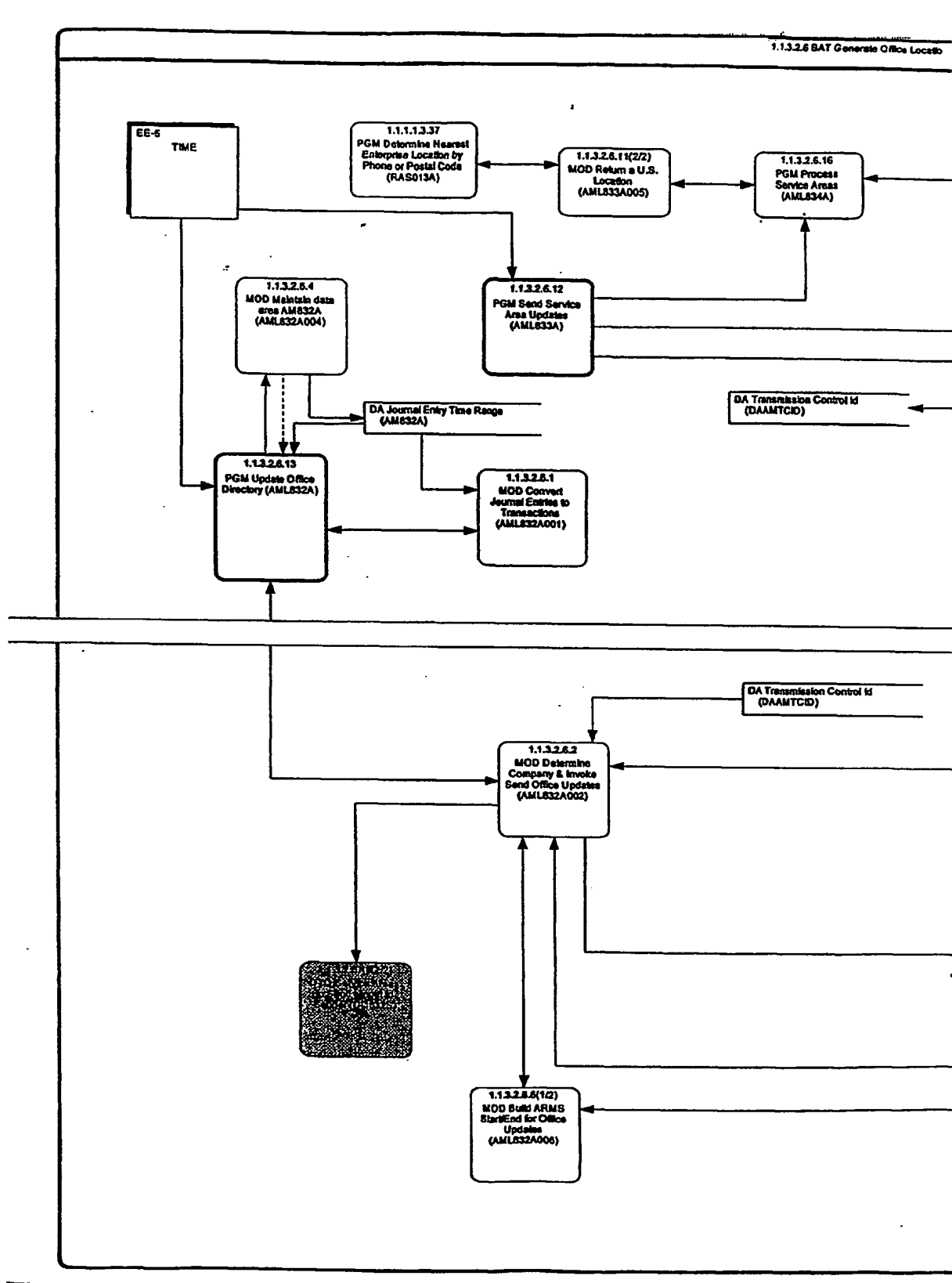
1.1.3.2.\*  
BT Synchronize  
Office Information for  
X12 Customer

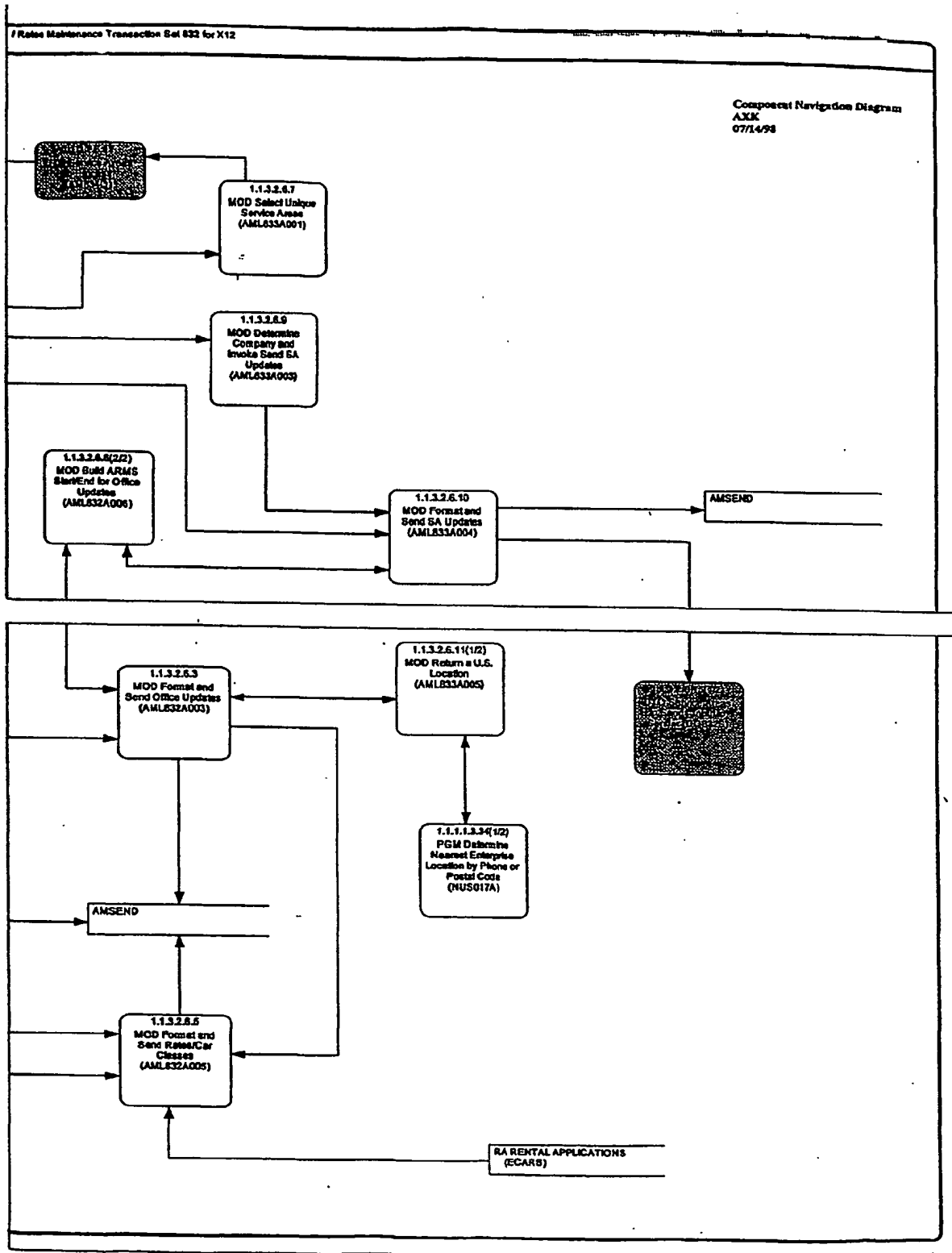
1.1.3.3.\*  
BT Send Initial or  
Yearly Car Class  
Rates and Initial  
Office Information for  
X12 Customer





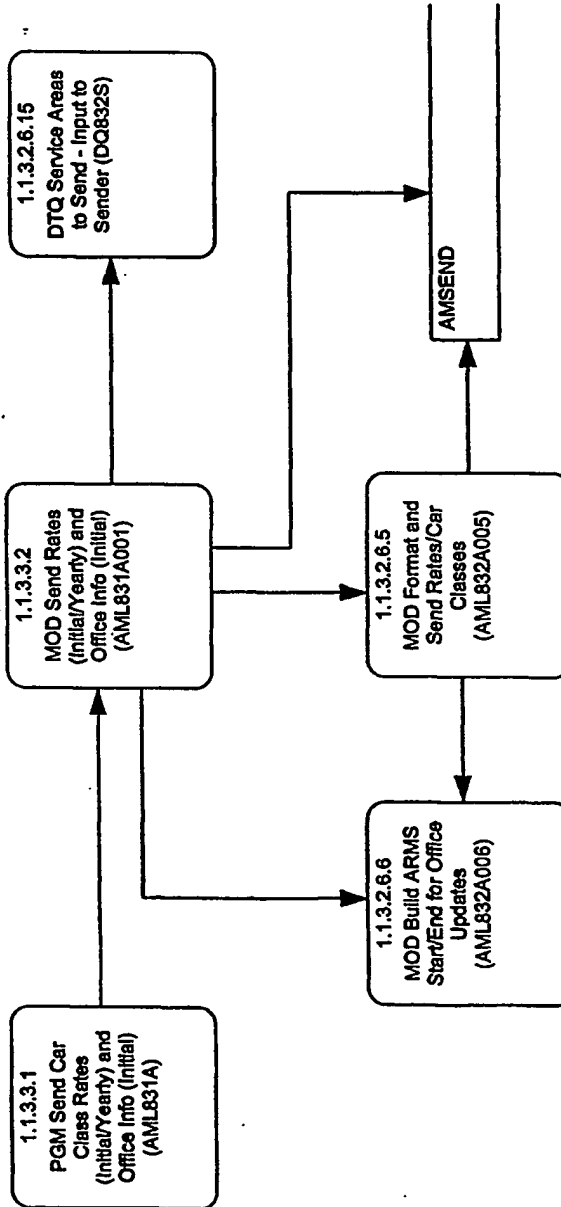




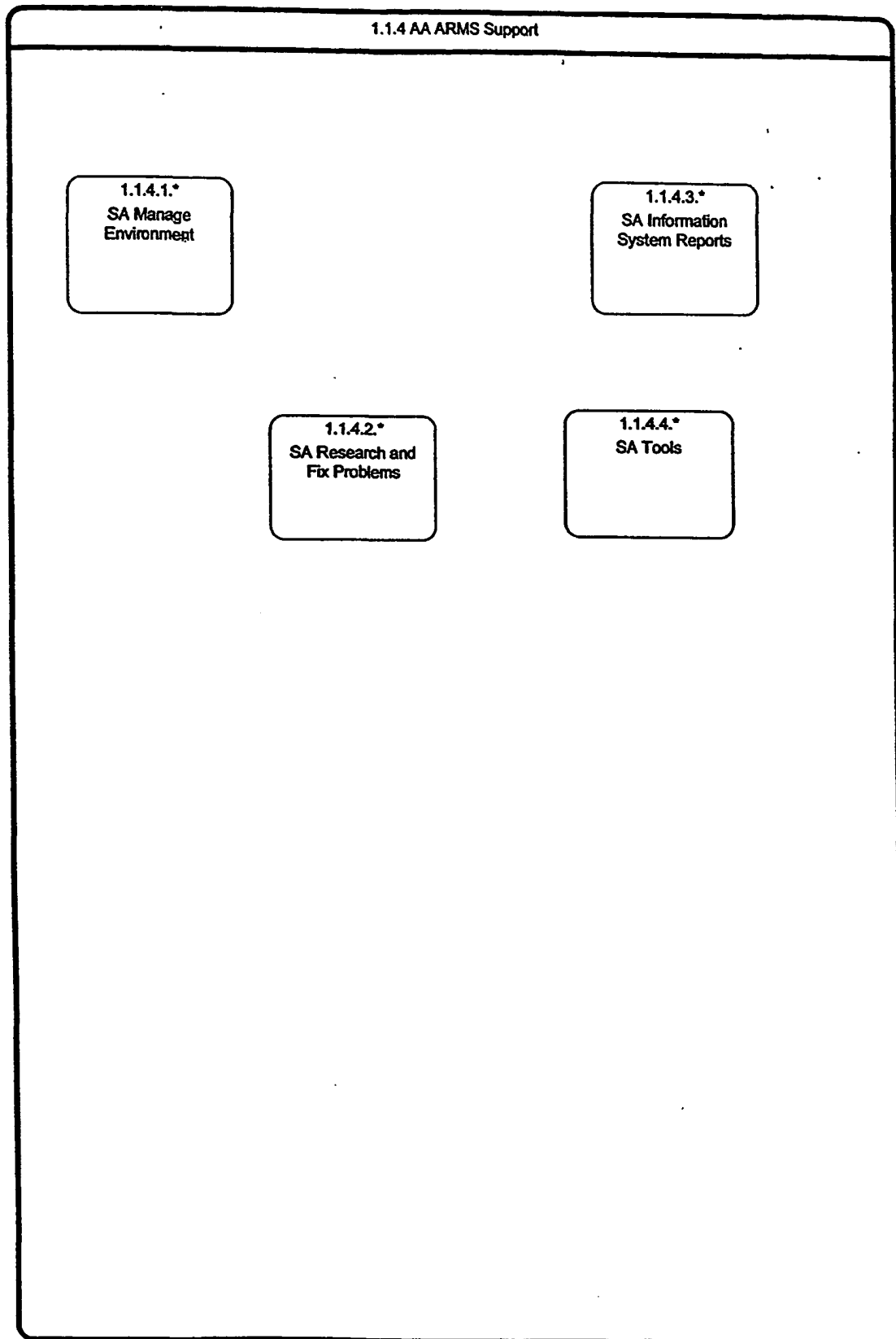


1.1.3.3 BT Send Initial or Yearly Car Class Rates and Initial Office Information for X12 Customer

Component Navigation  
Diagram  
ALM, RDS  
7/14/99

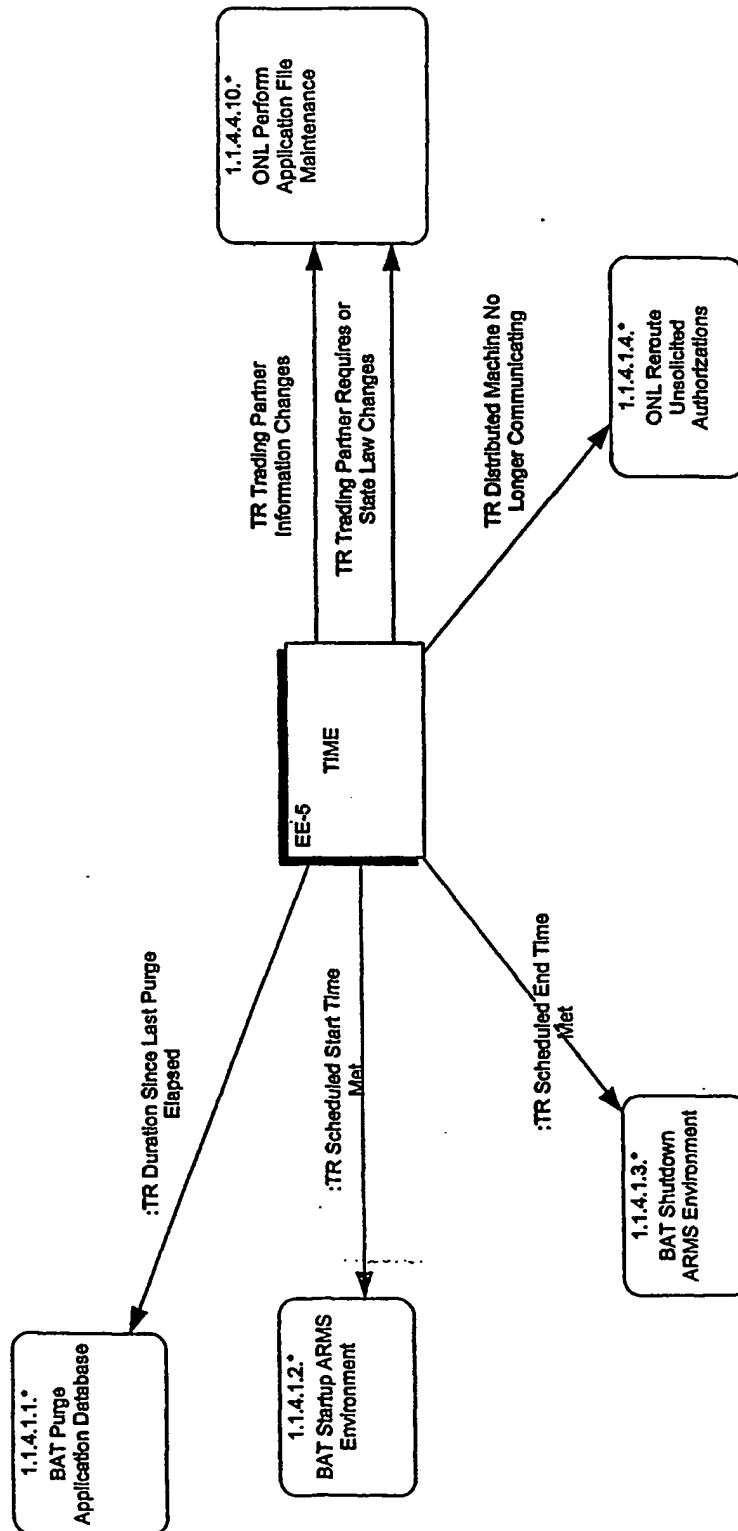


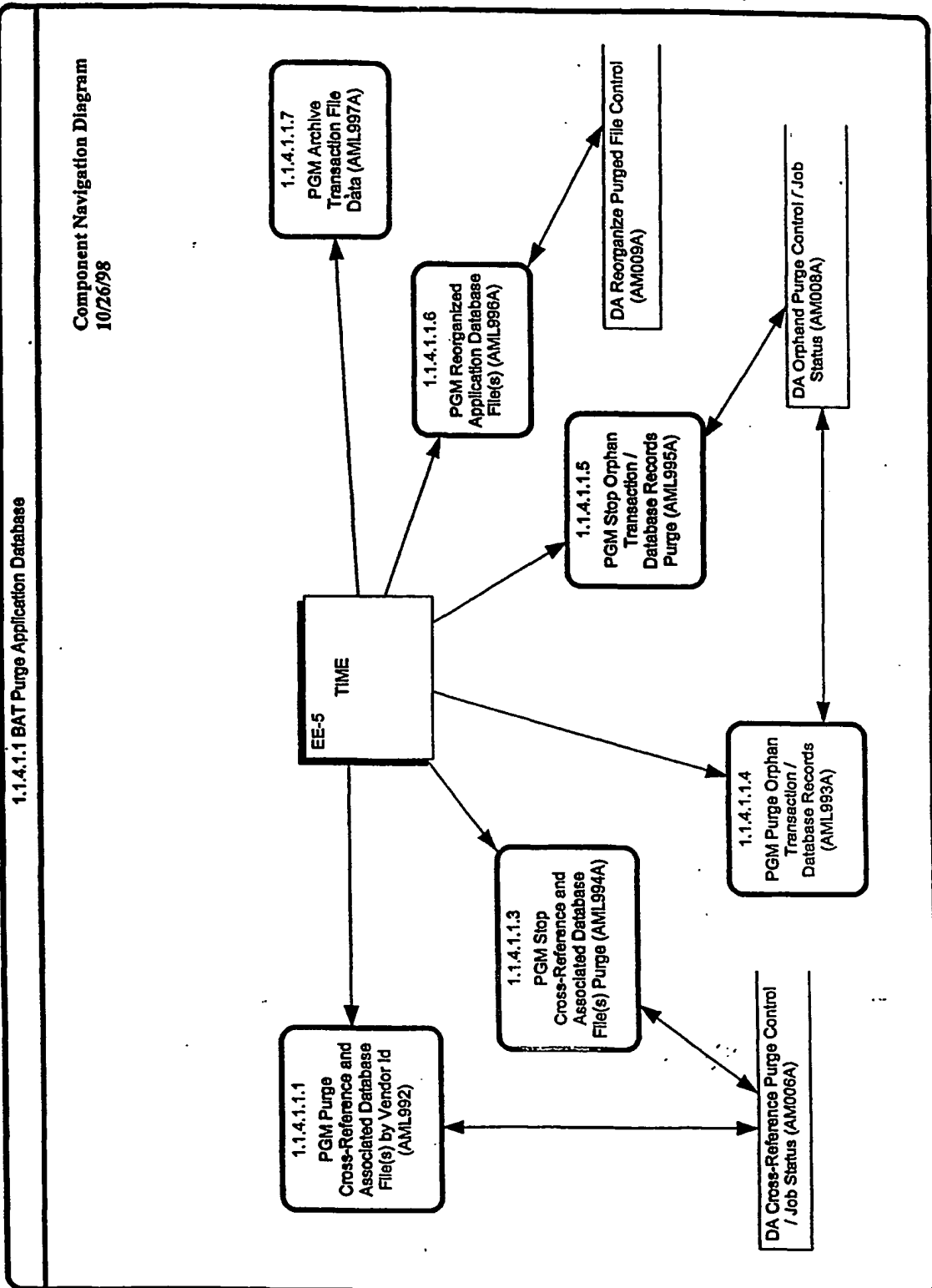


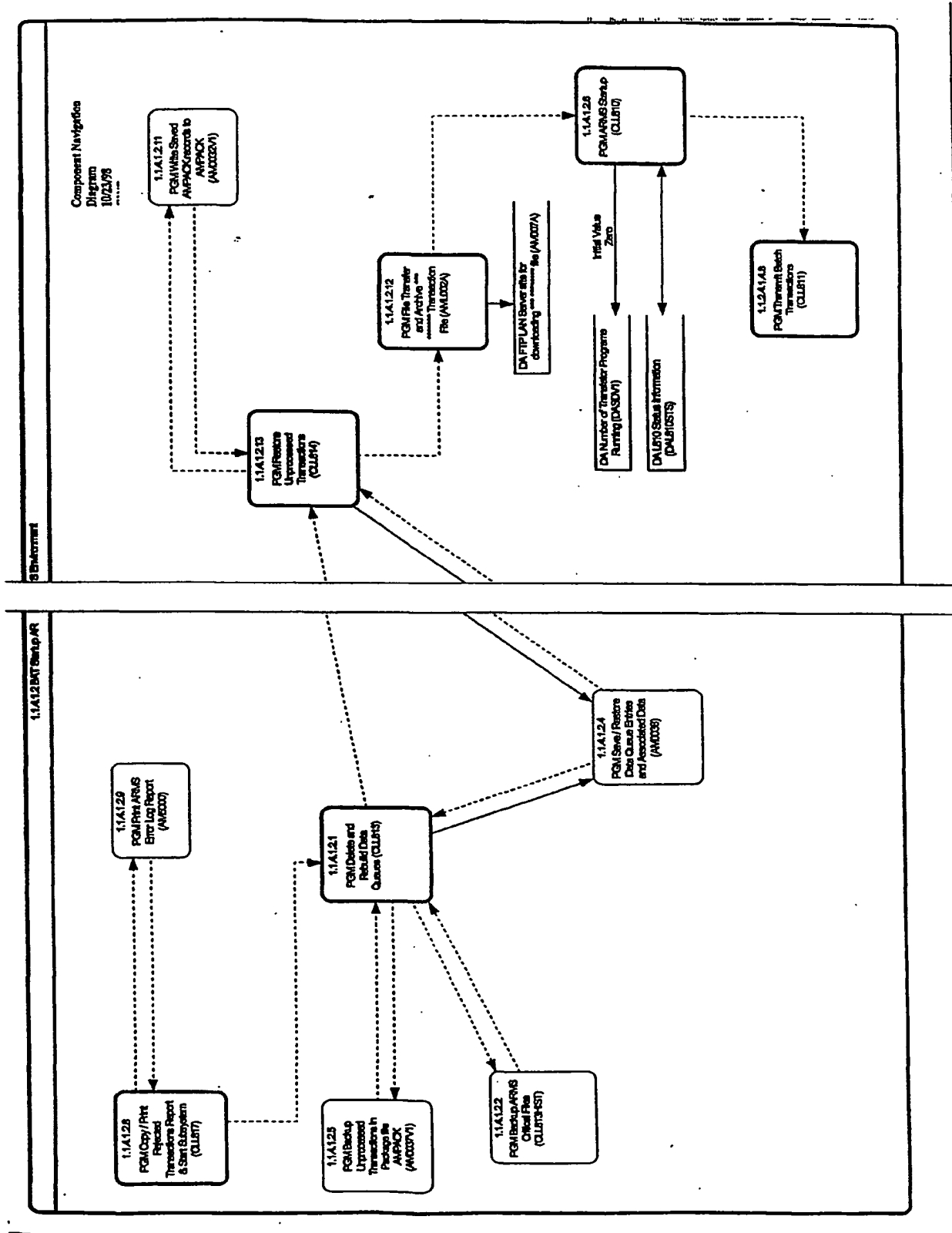


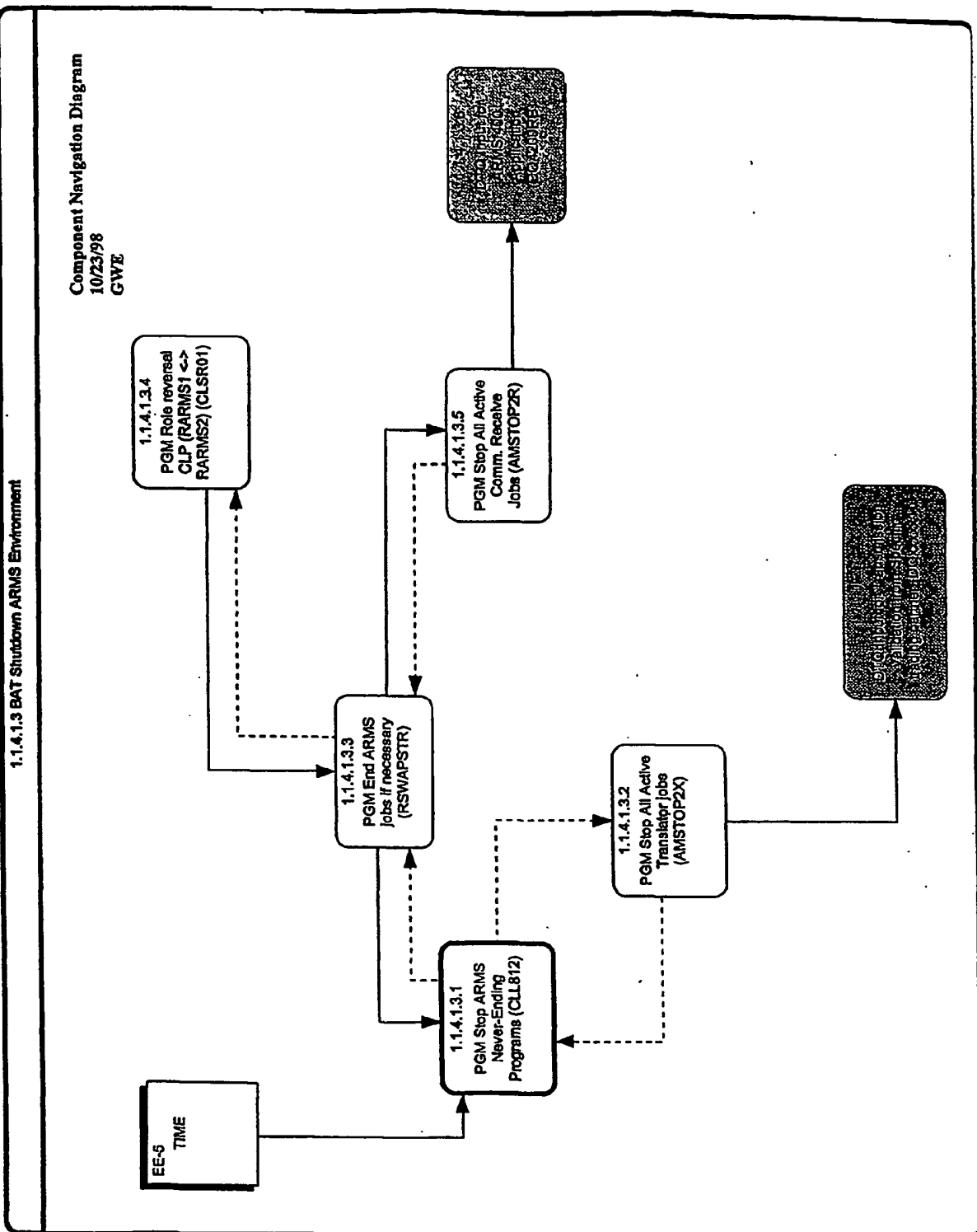
1.1.4.1 SA Manage Environment

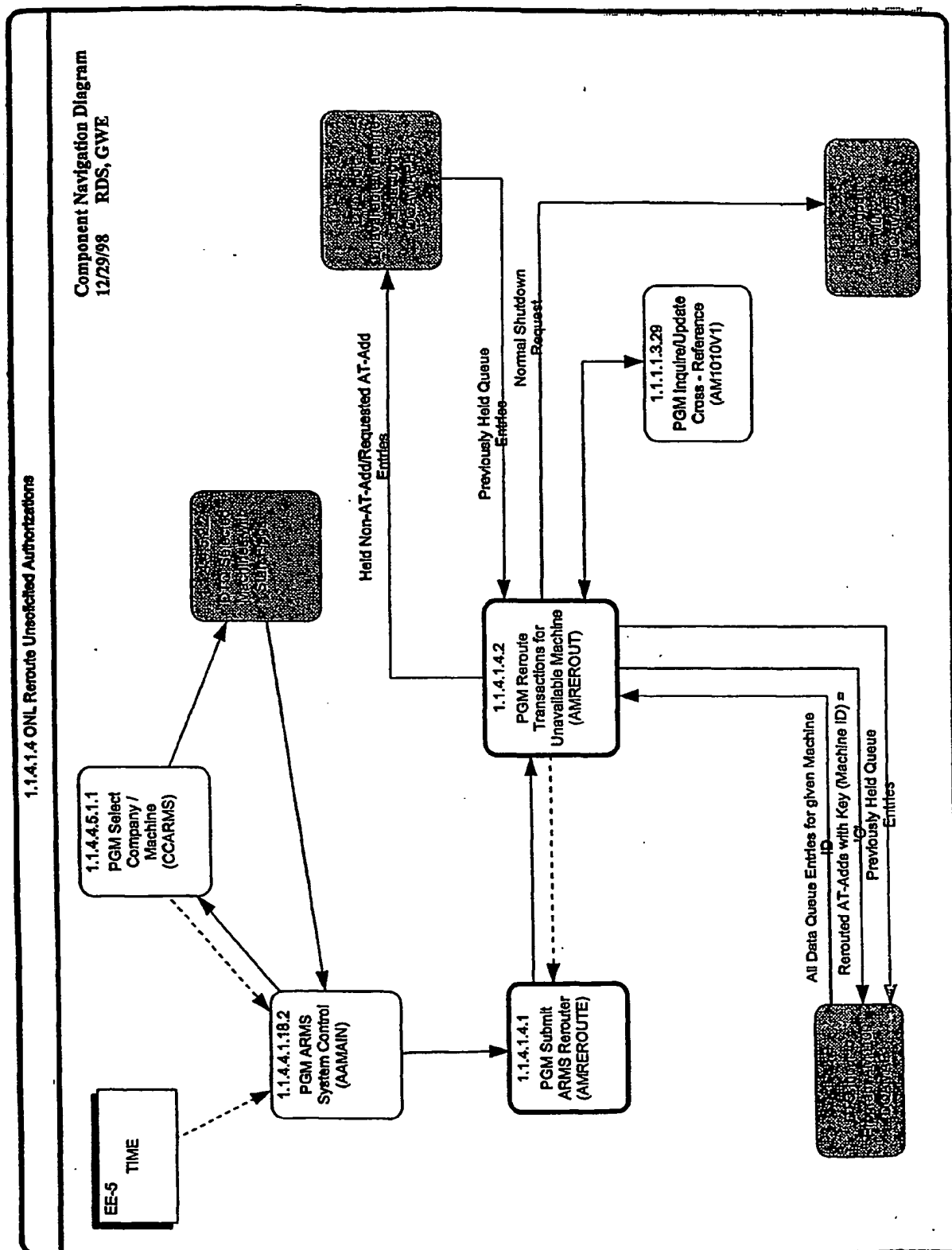
Activity Dependency Diagram  
01/27/99 GWE

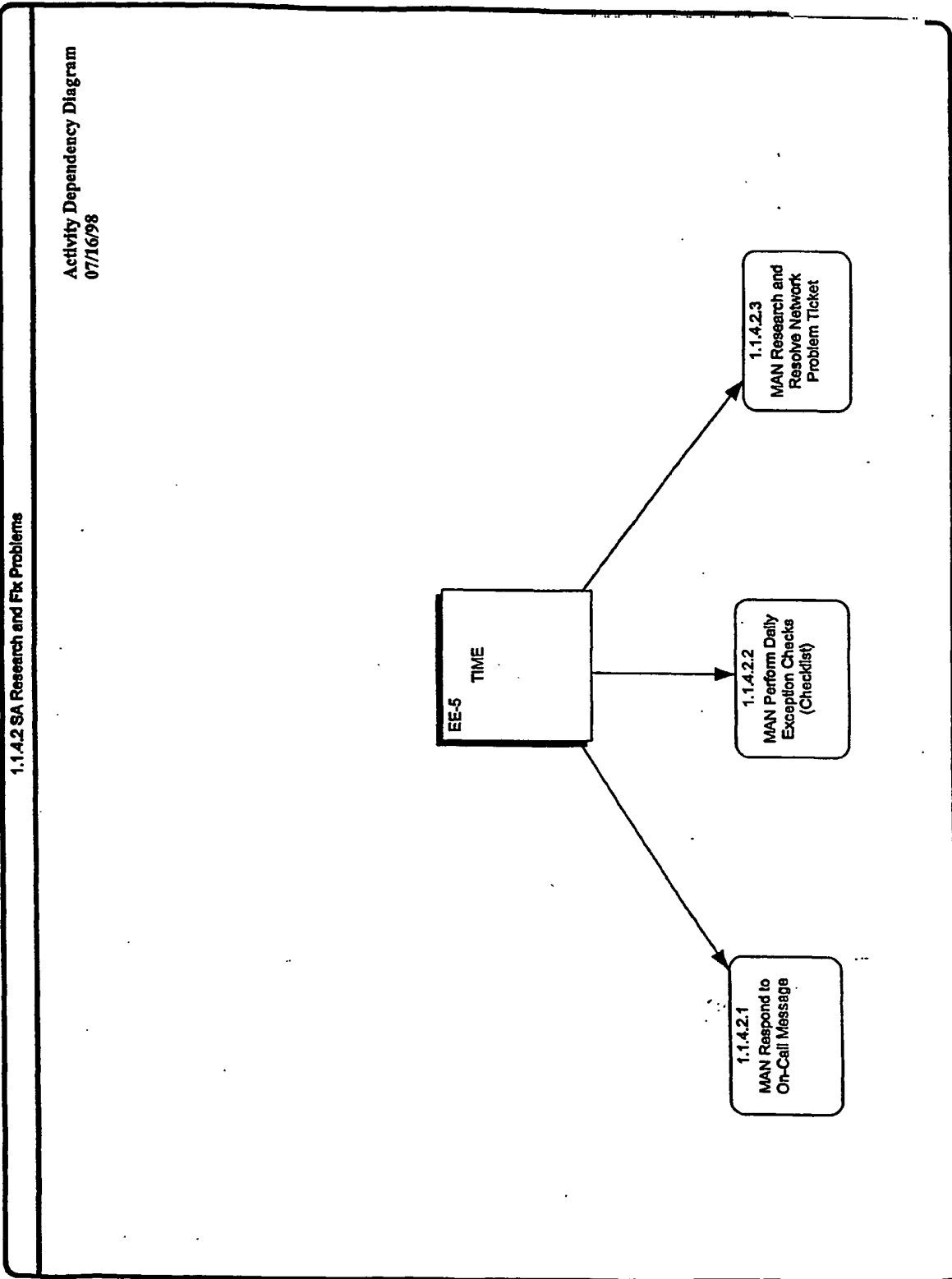


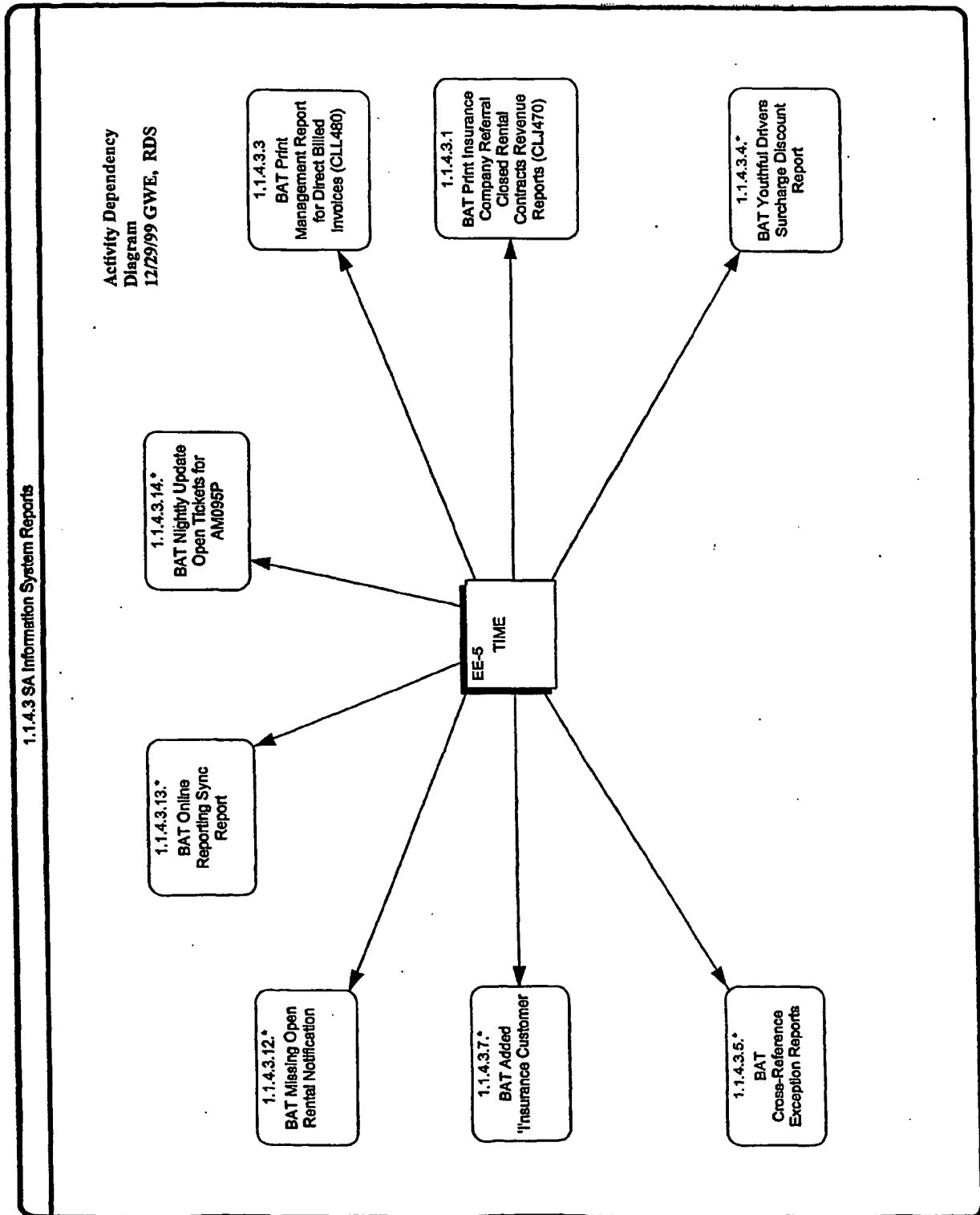




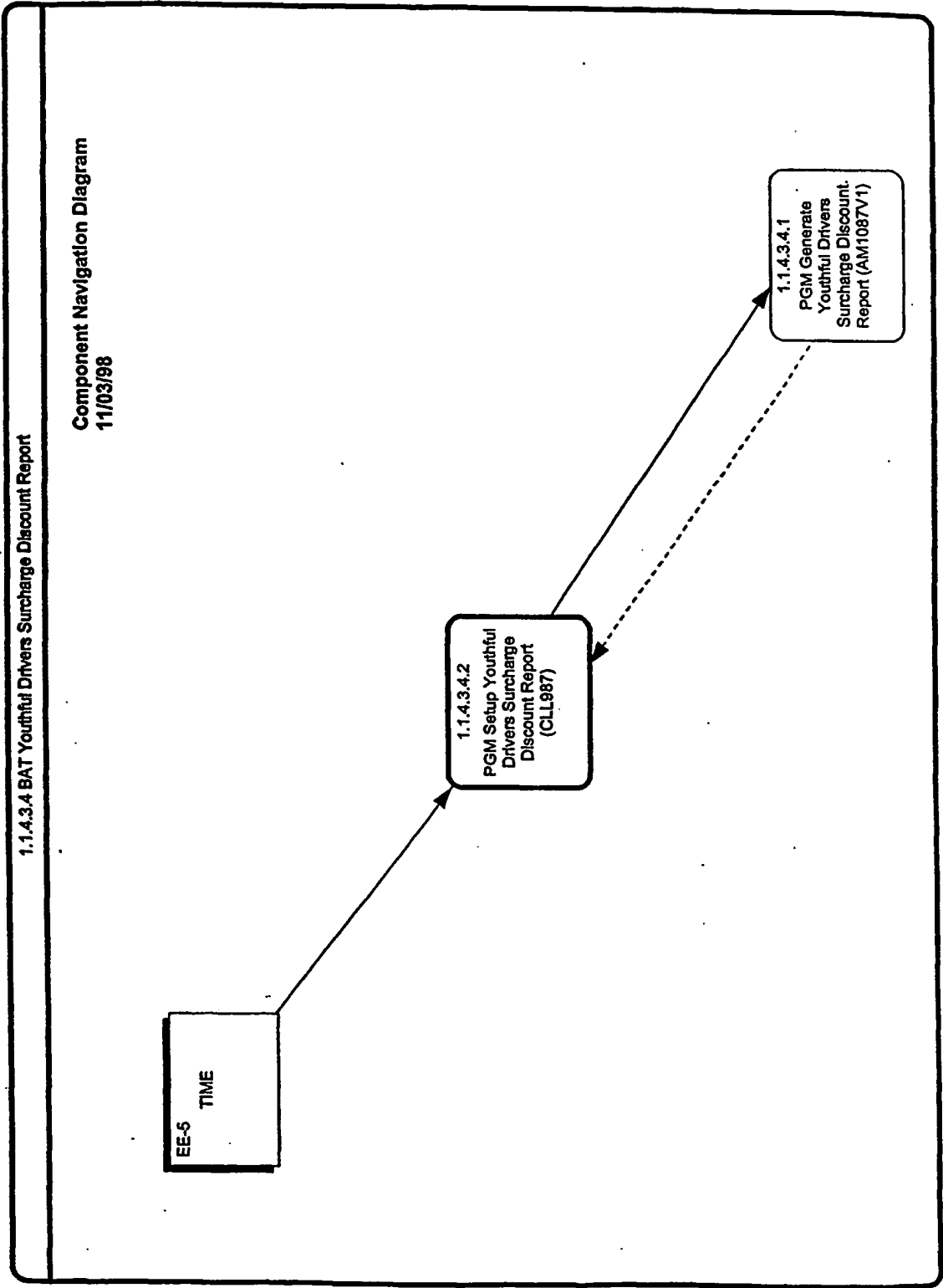


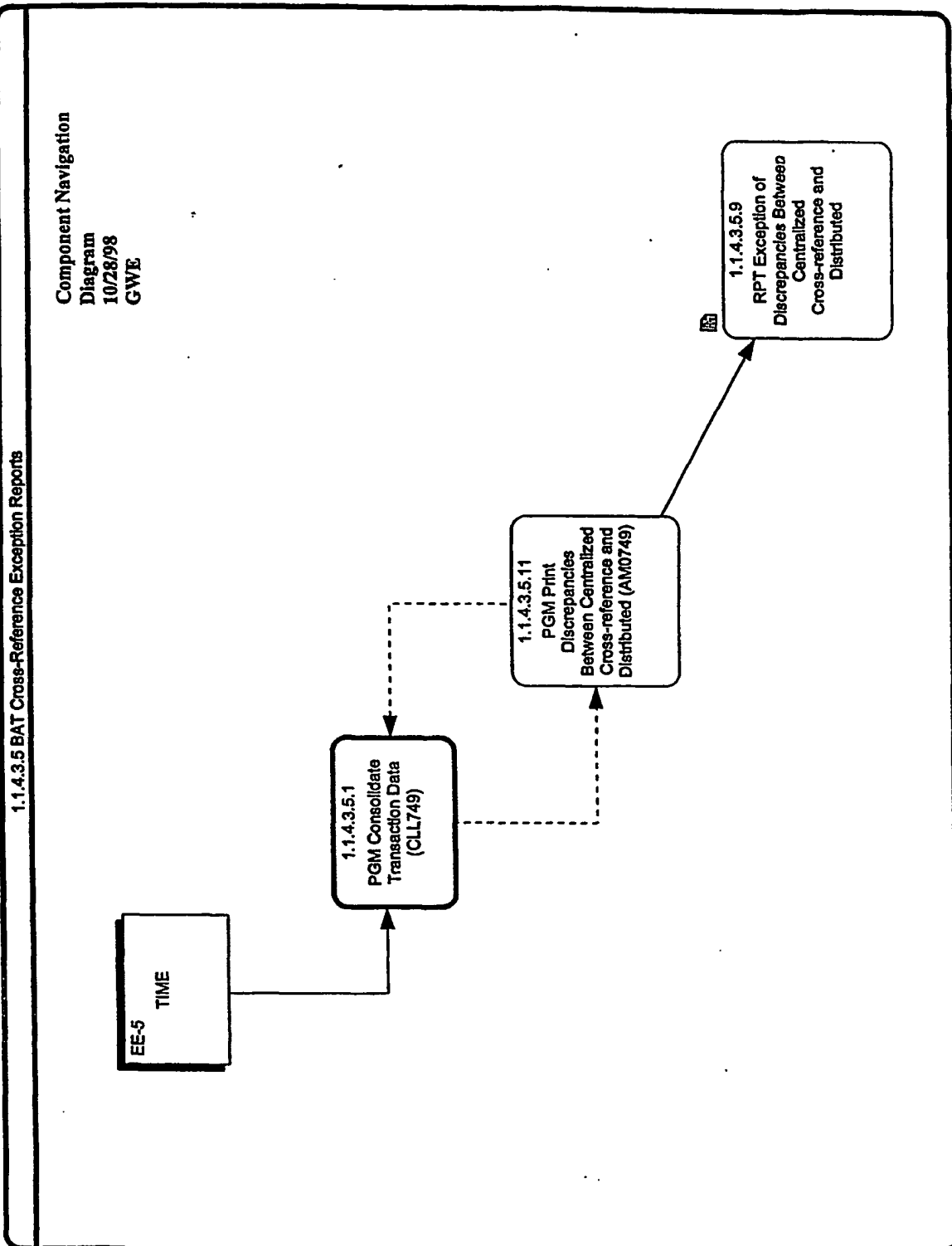


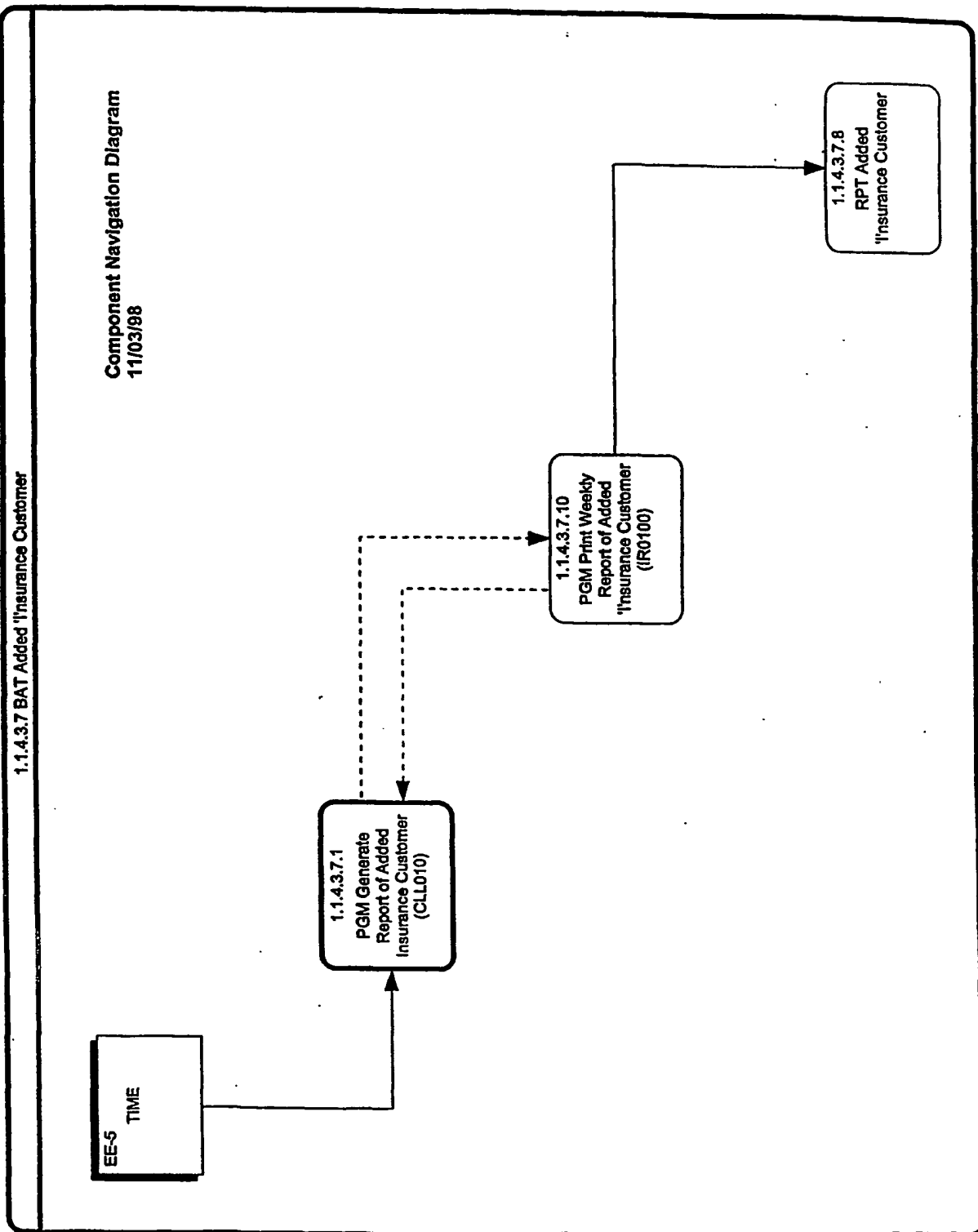


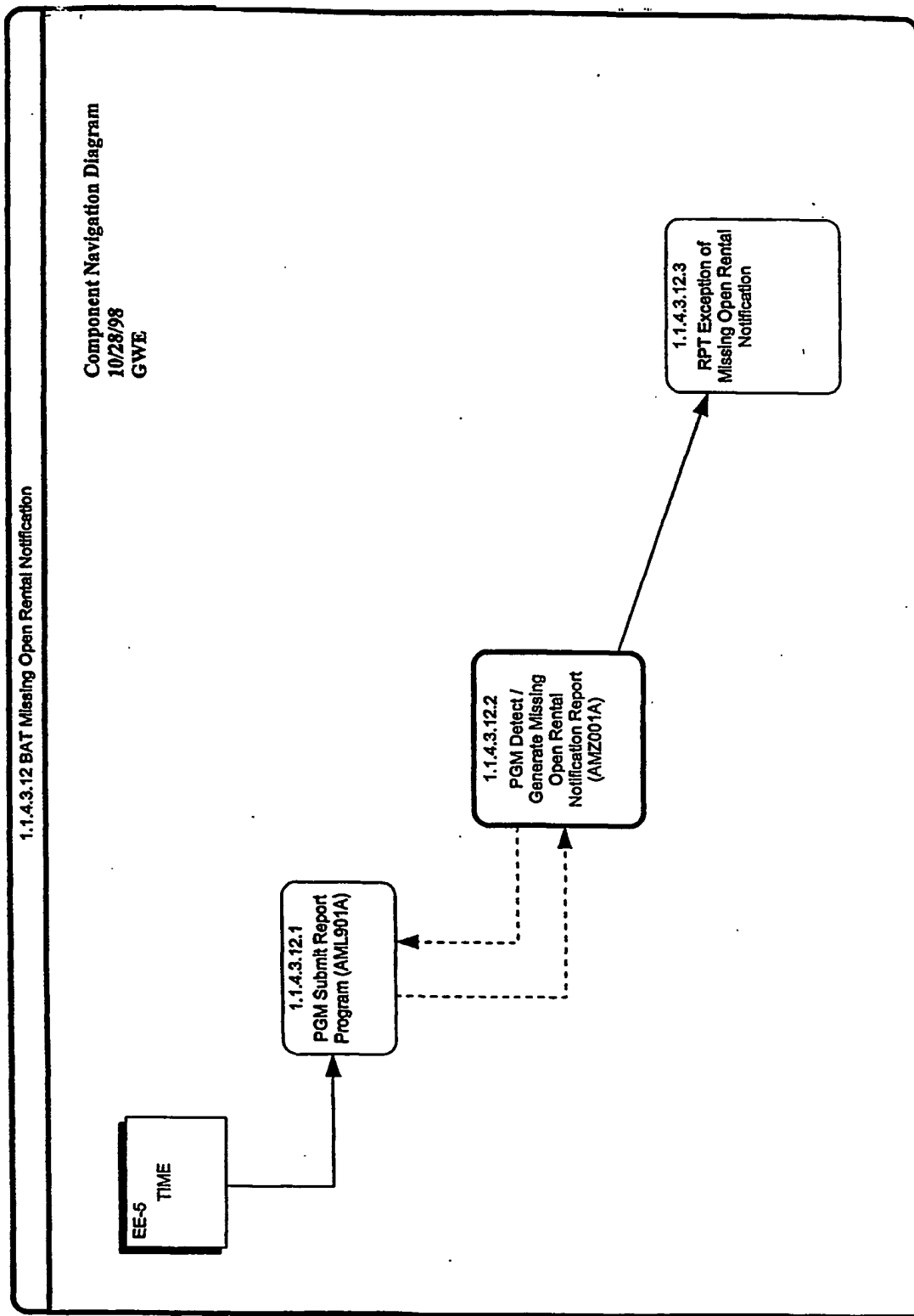


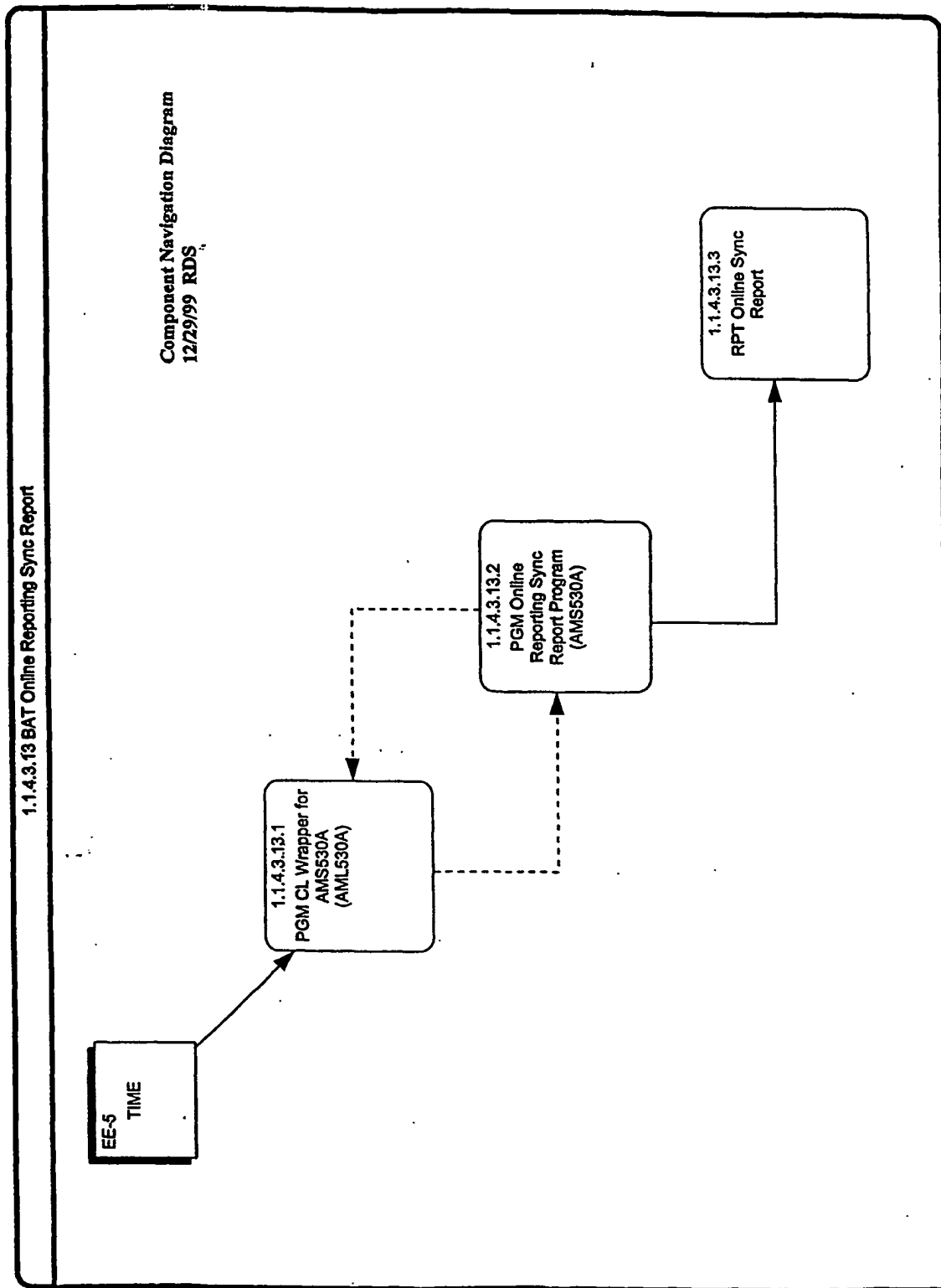






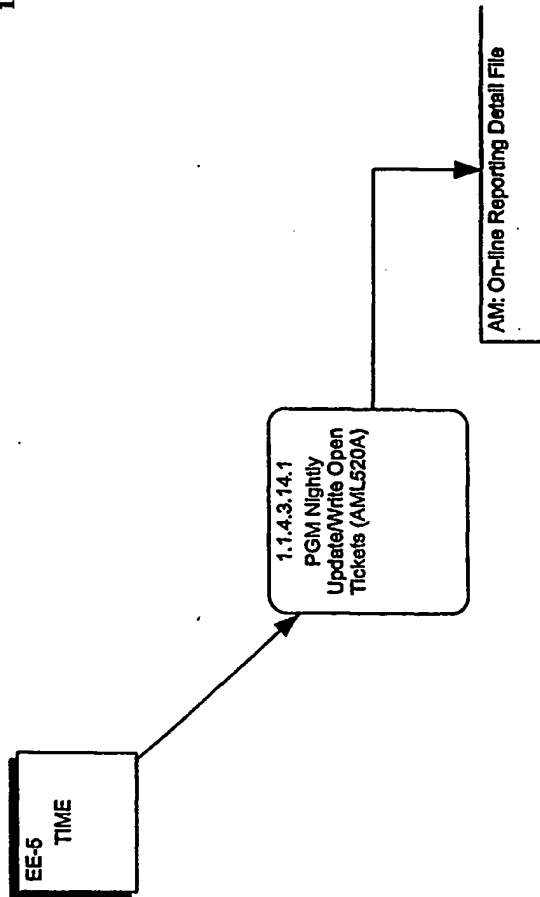


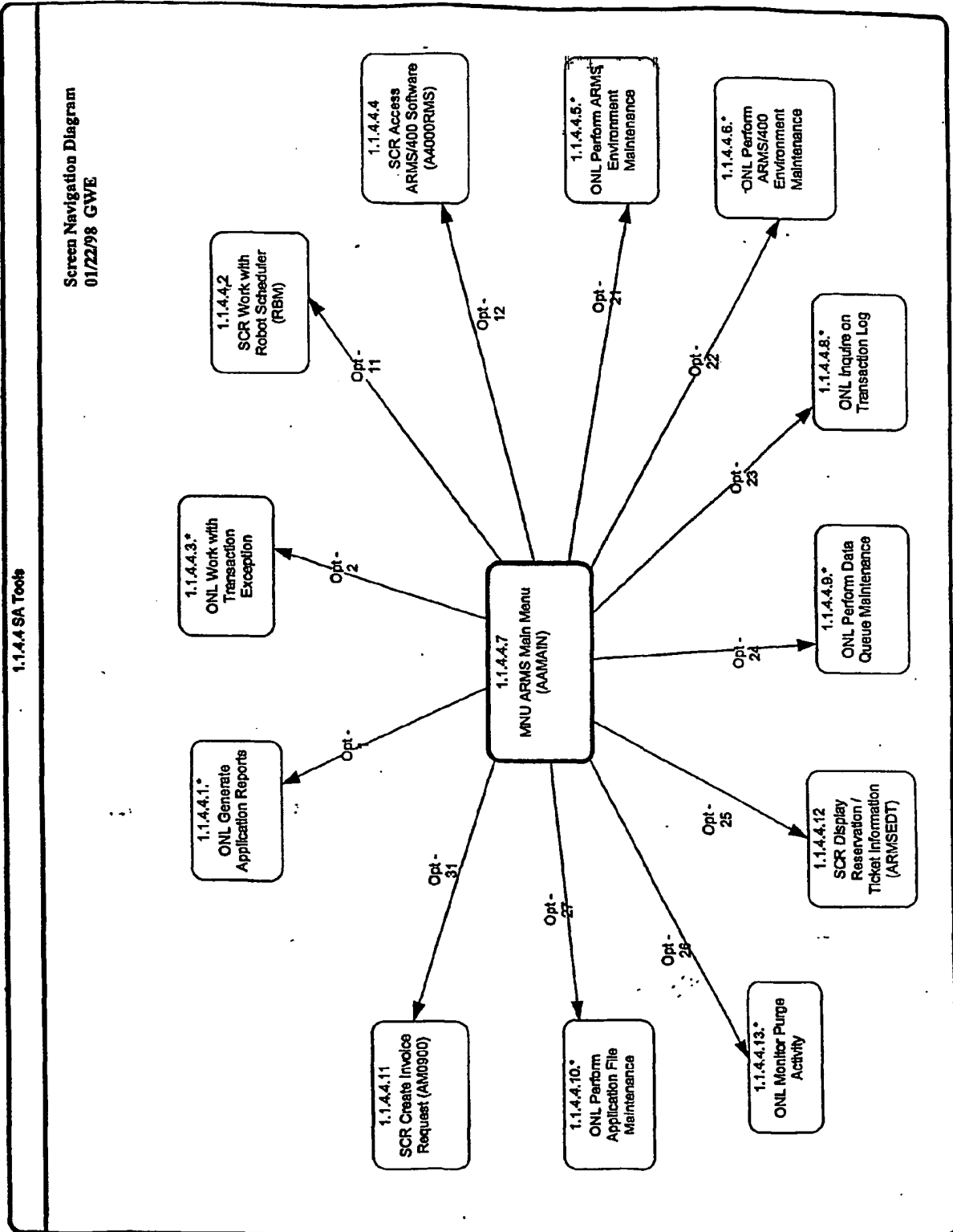


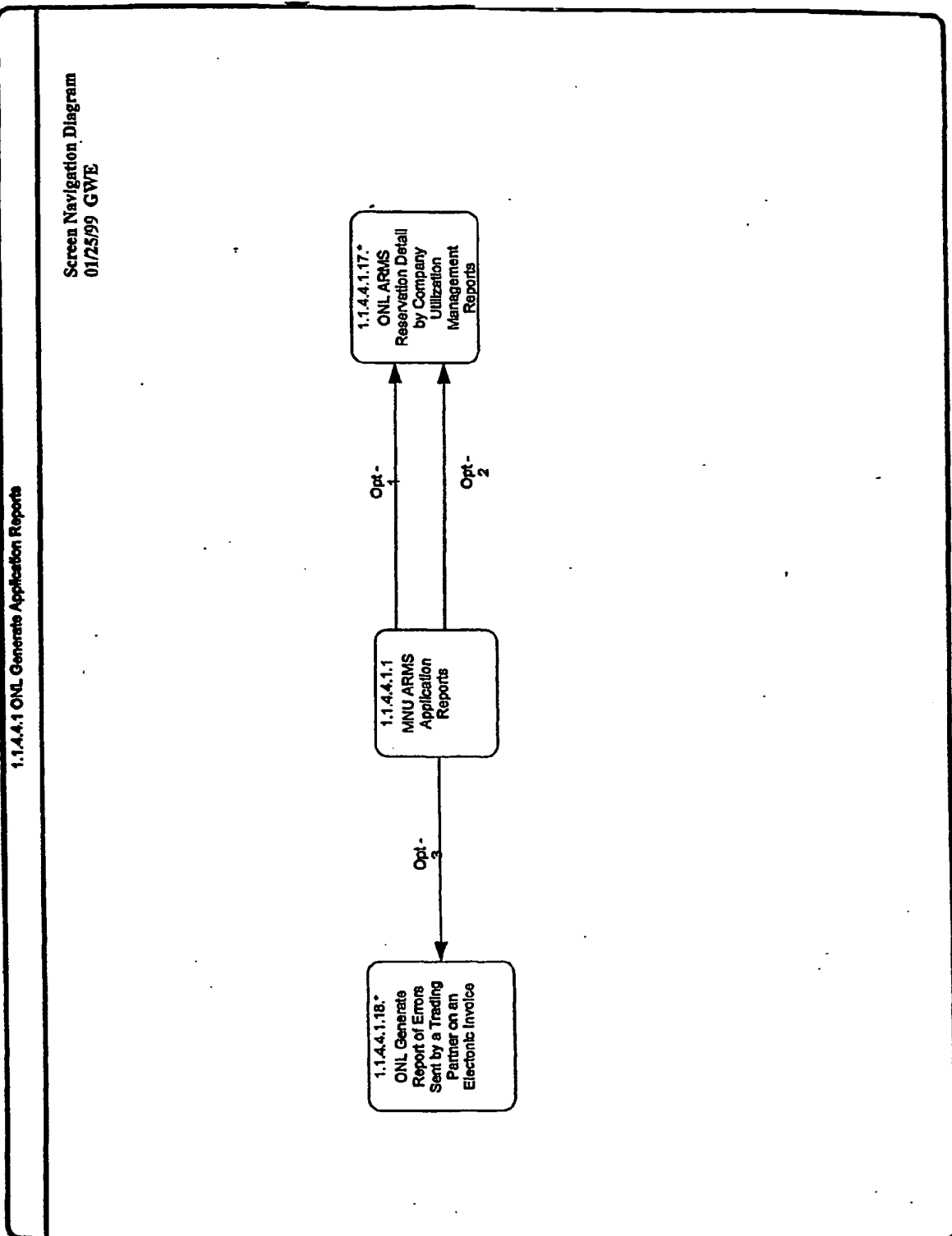


1.1.4.3.14 BAT Nightly Update Open Tickets for AM095P

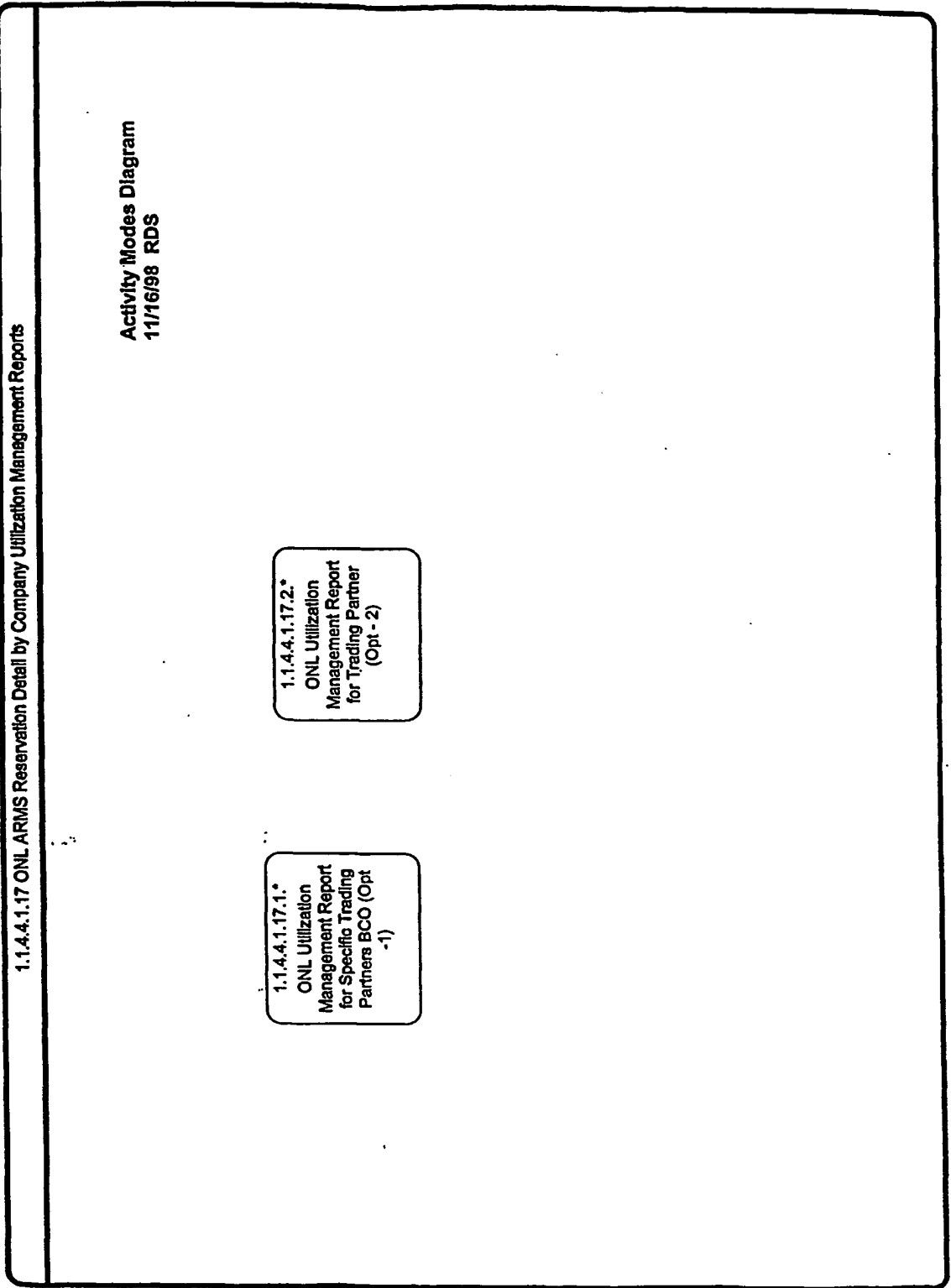
Component Navigation  
Diagram  
12/29/99 RDS





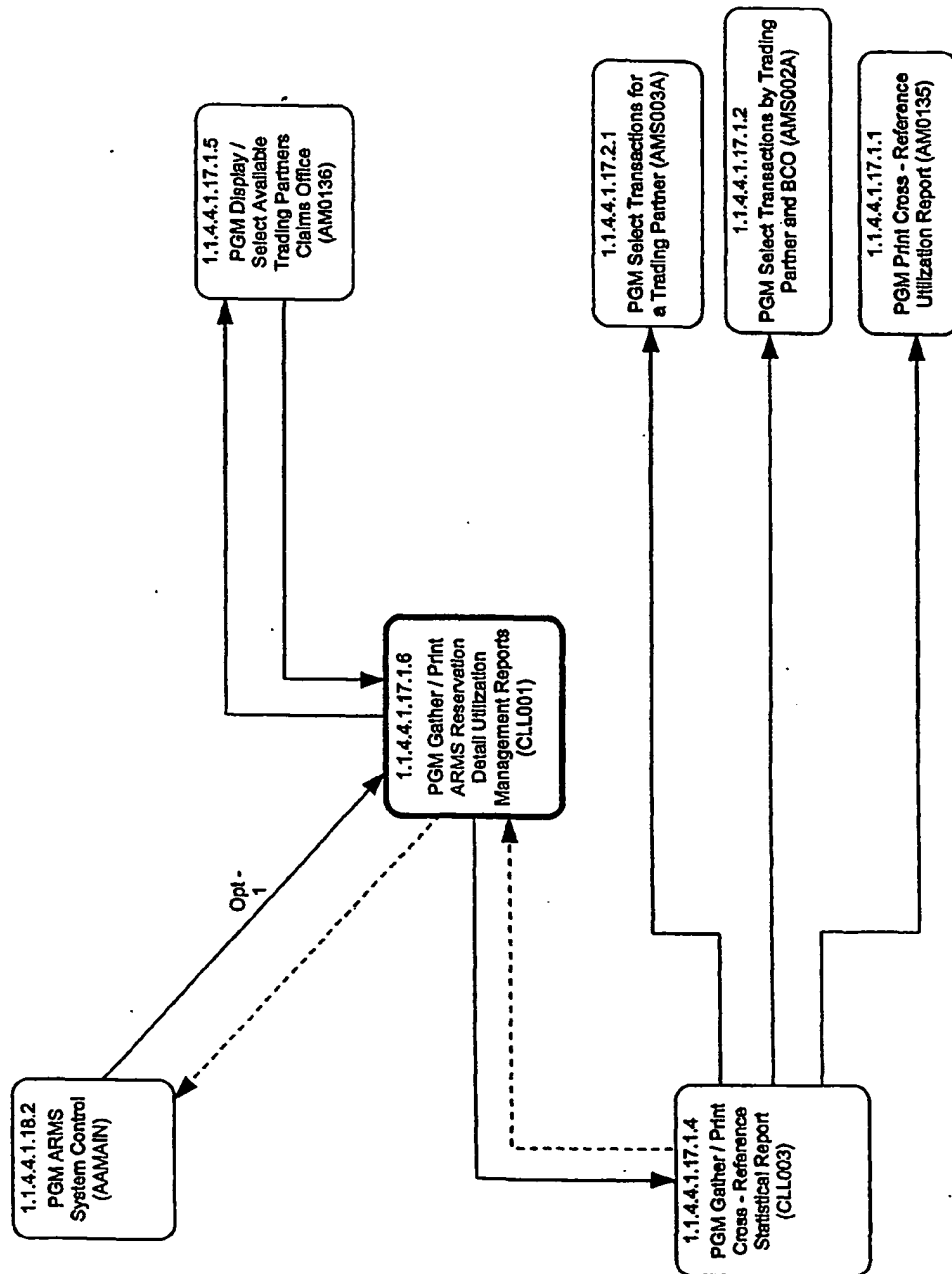


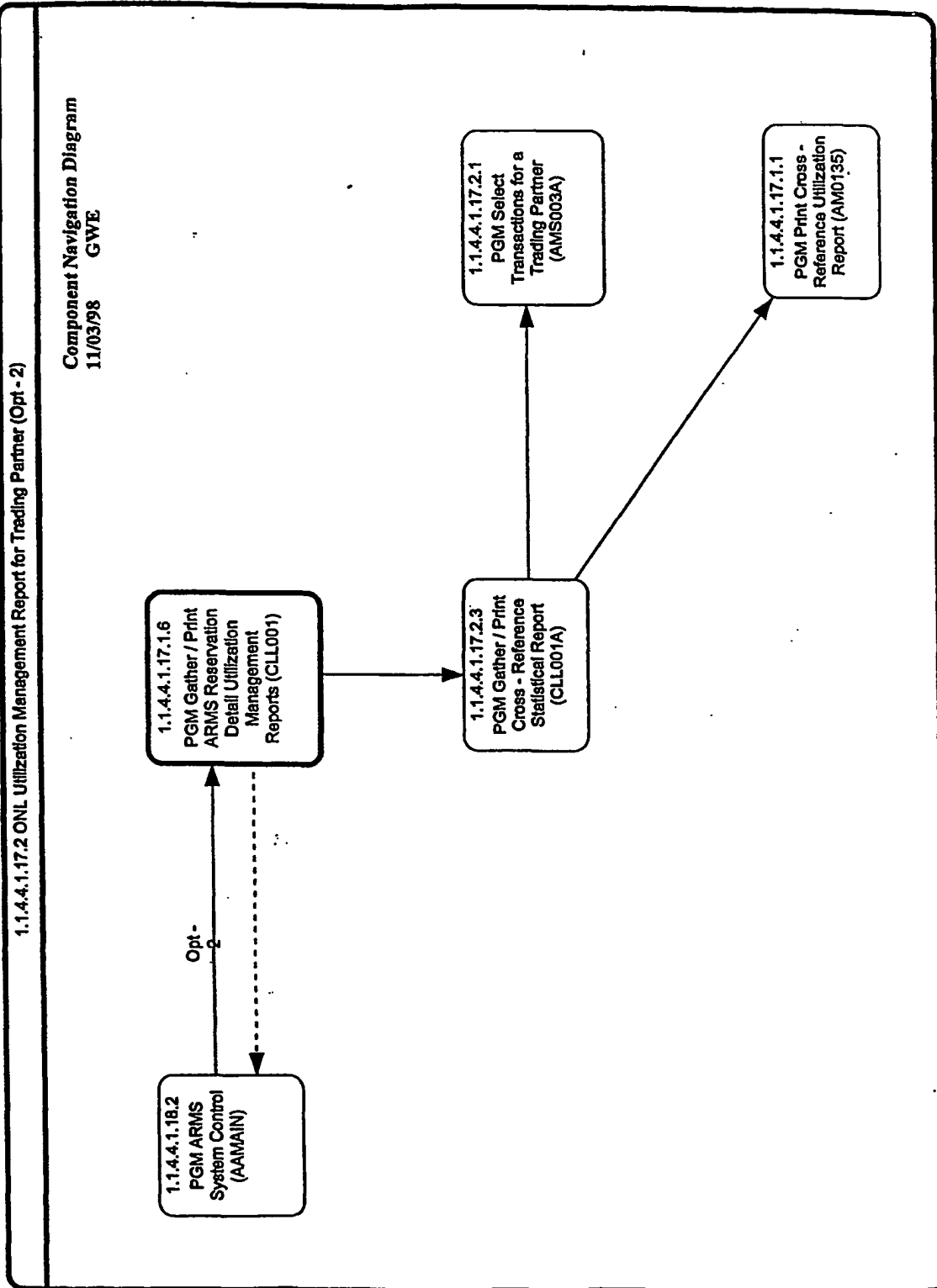




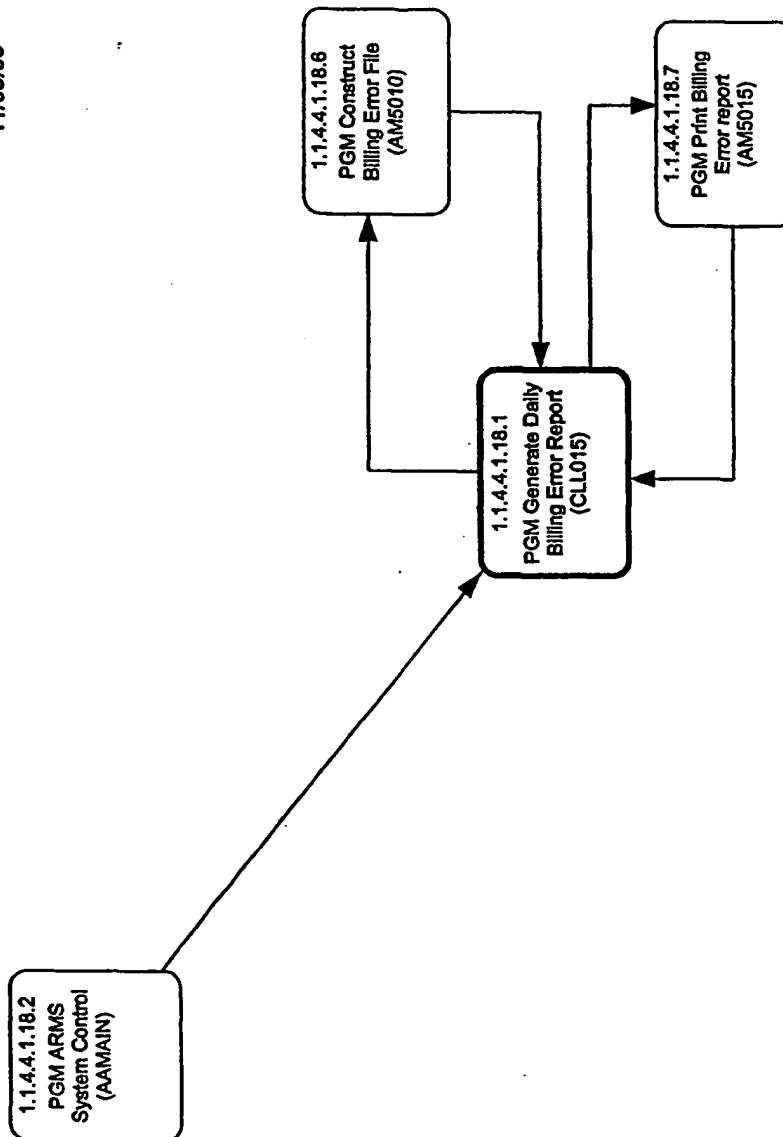
## 1.1.4.4.1.17.1 ONL Utilization Management Report for Specific Trading Partners BCO (Opt-1)

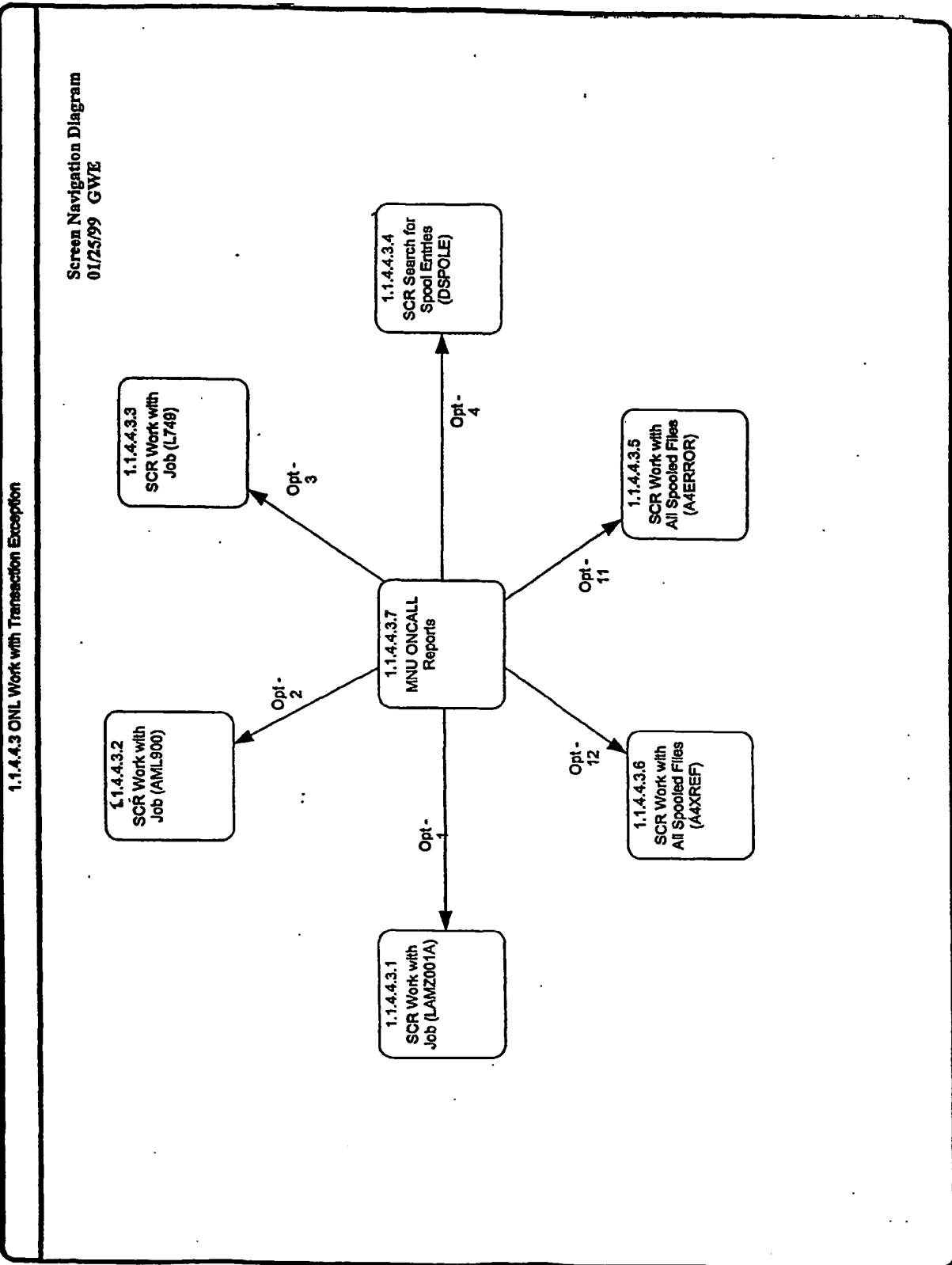
Component Navigation Diagram  
11/02/98 GWE

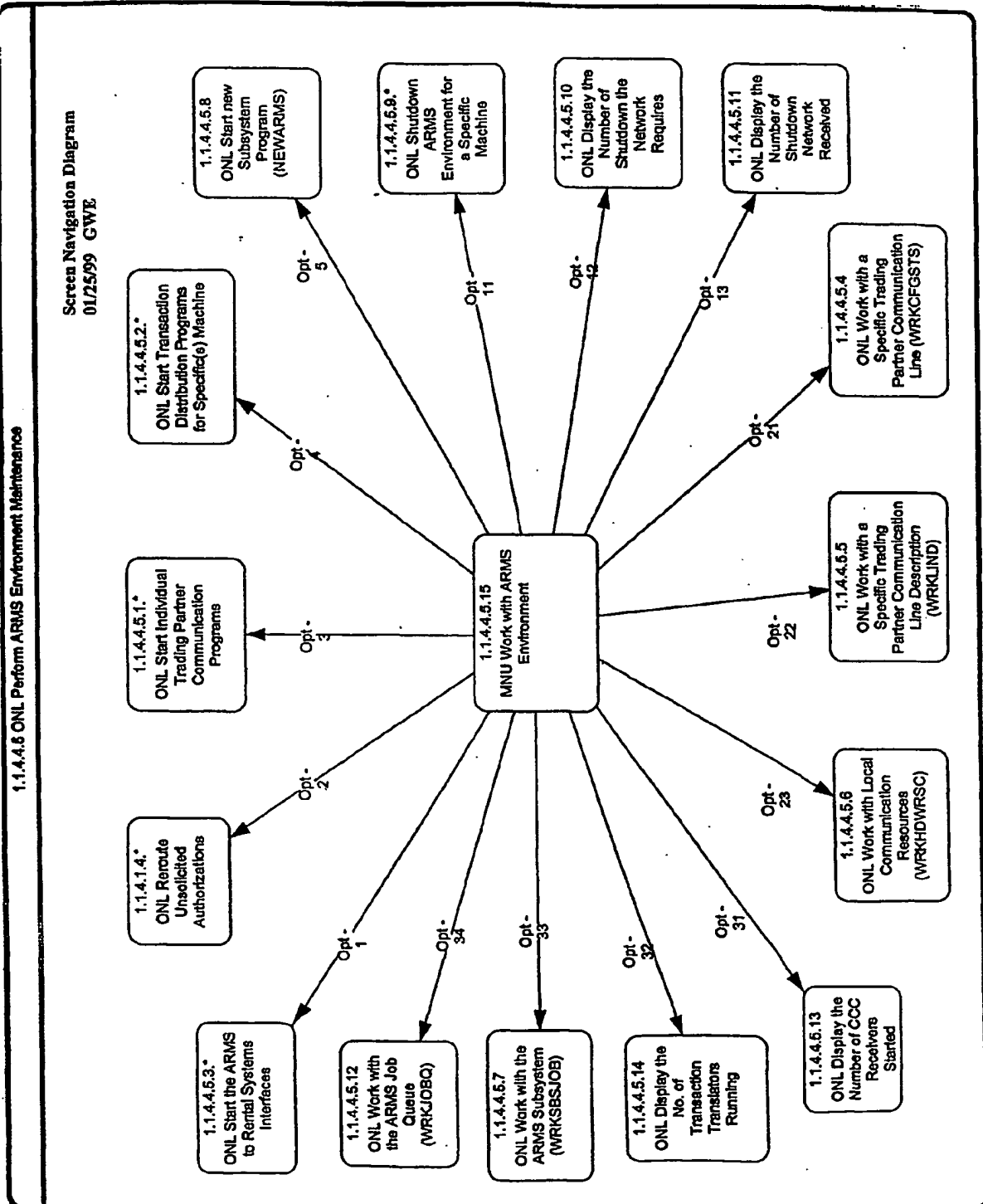


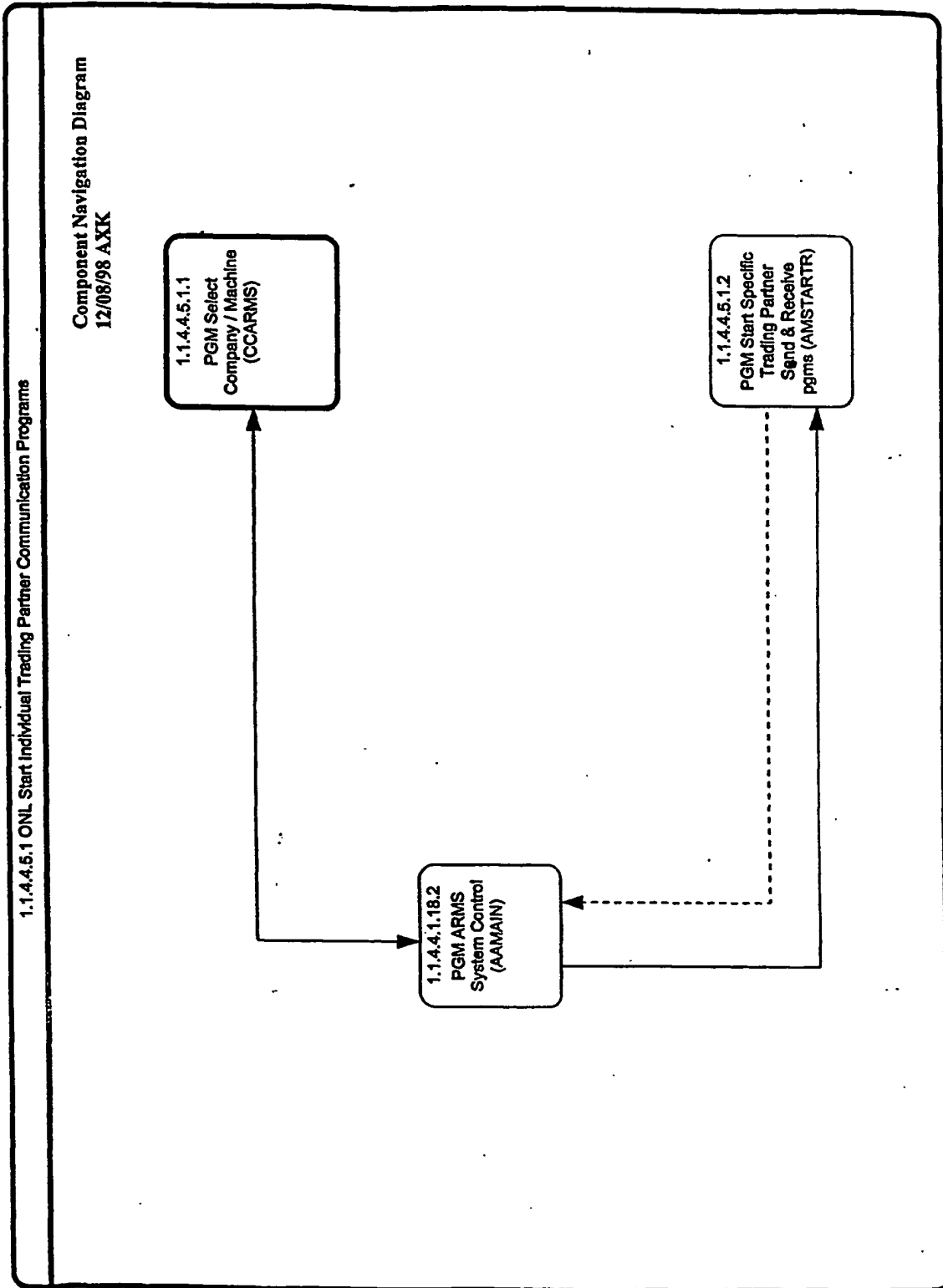


## 1.1.4.4.1.18 ONL Generate Report of Errors Sent by a Trading Partner on an Electronic Invoice

Component Navigation Diagram  
11/03/98

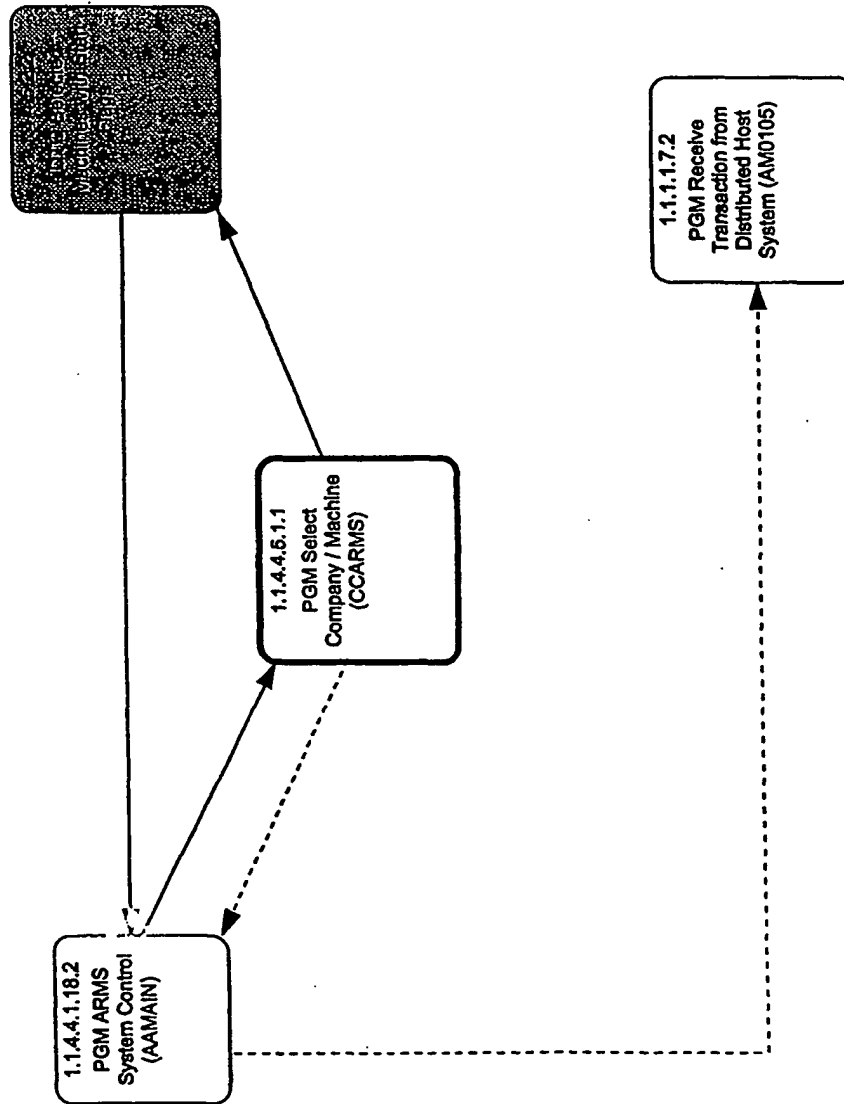




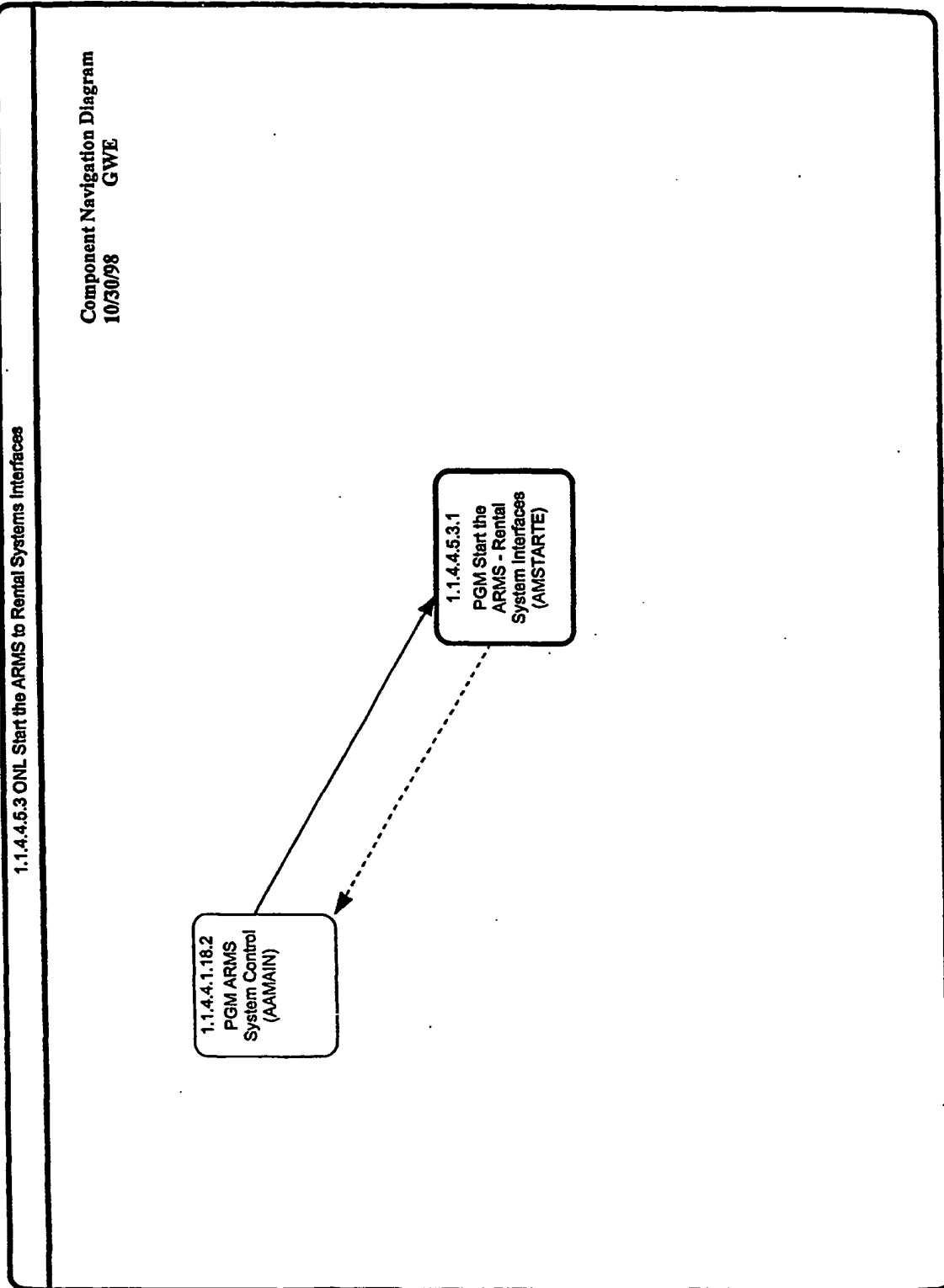


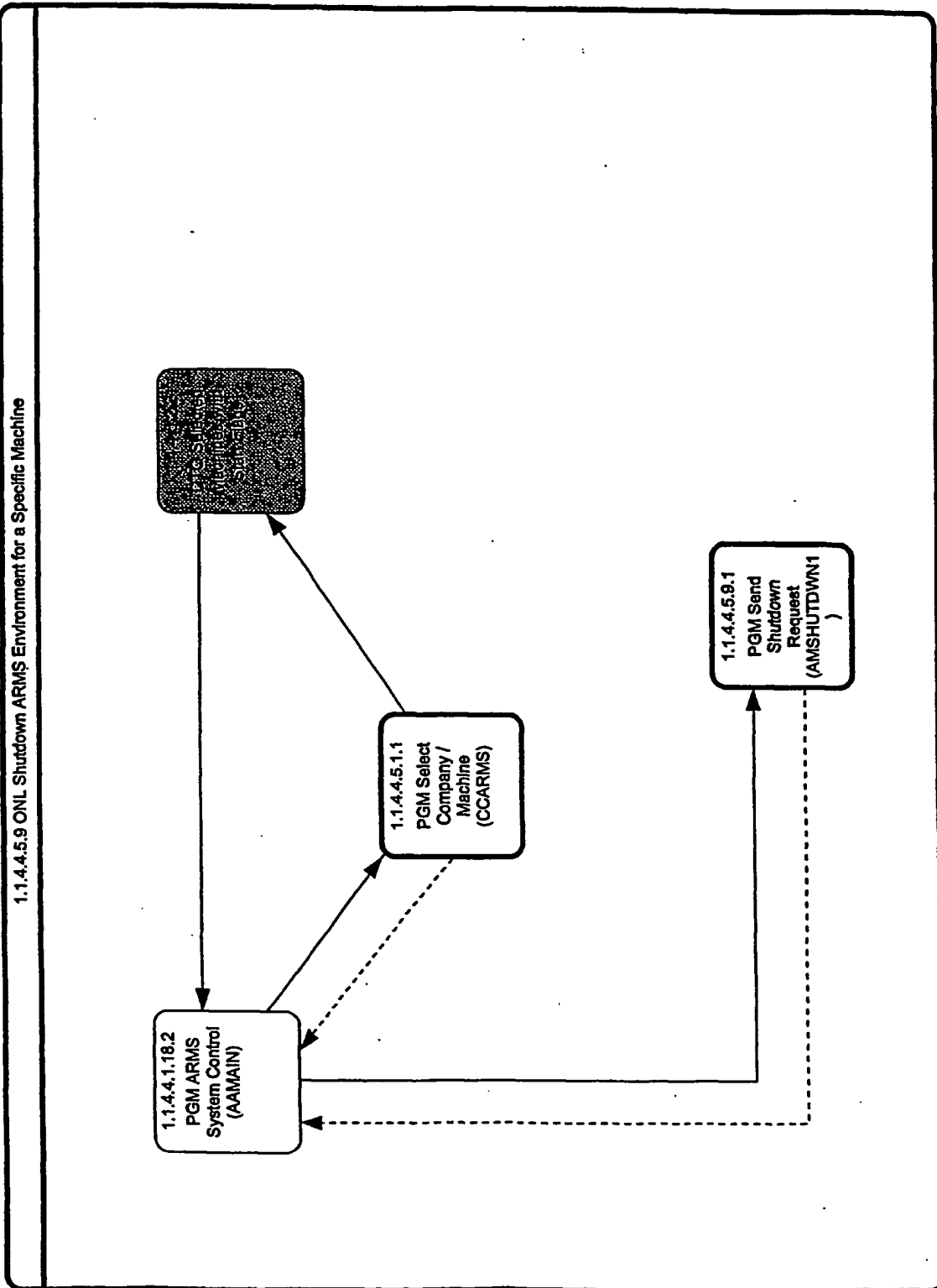
1.1.4.4.5.2 ONL Start Transaction Distribution Programs for Specific(s) Machine

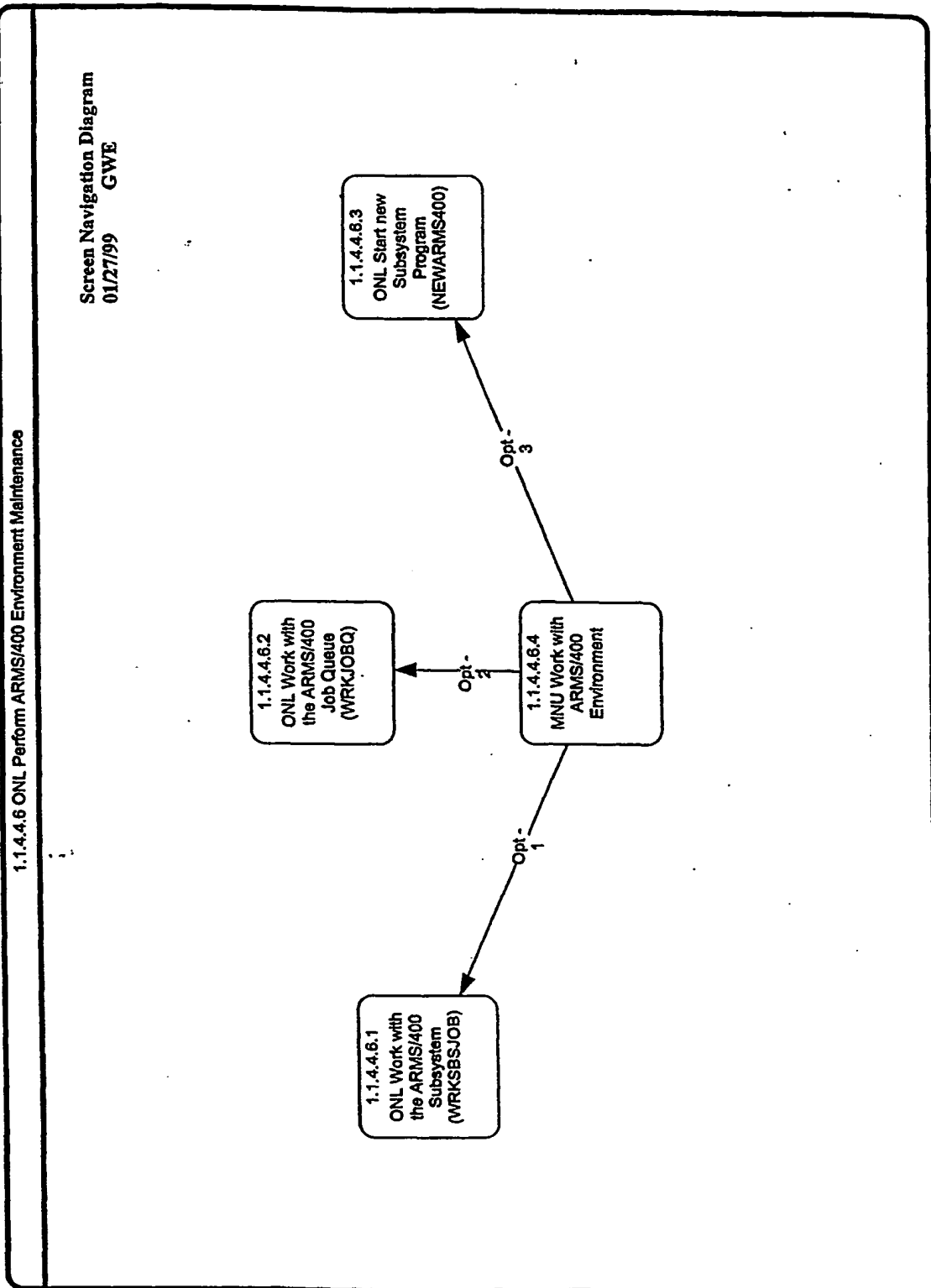
Component Navigation Diagram  
12/08/98 AXK





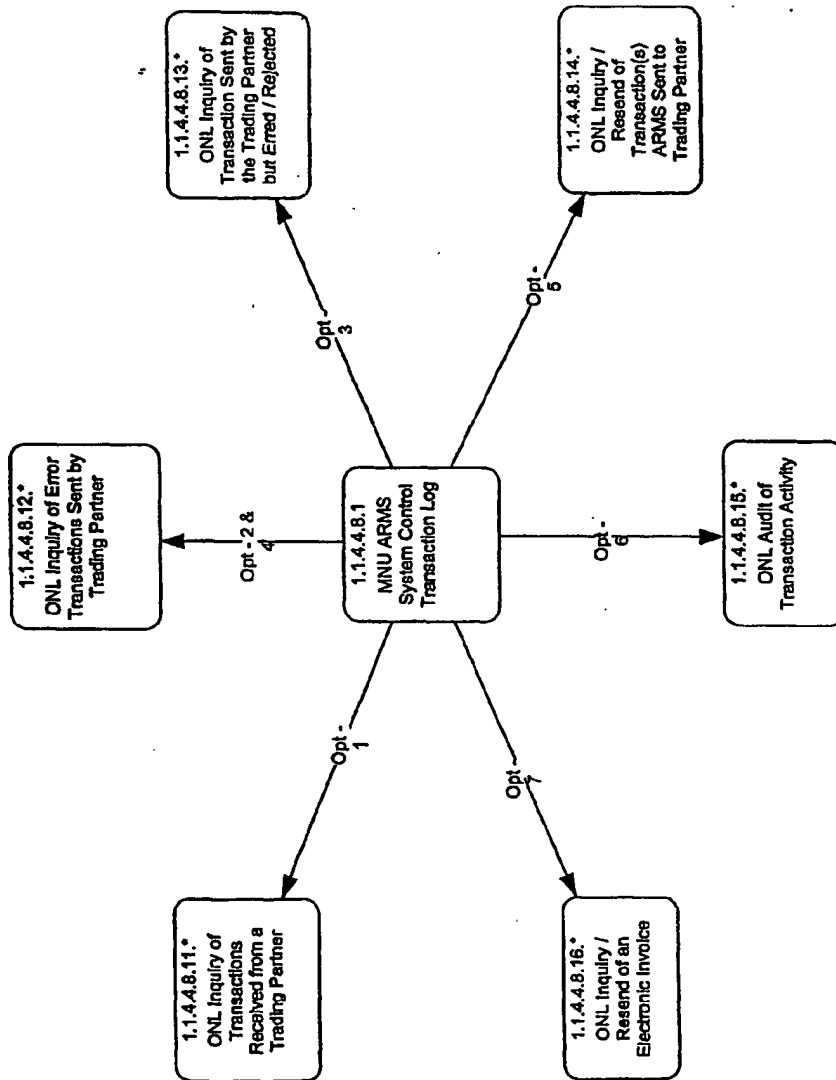


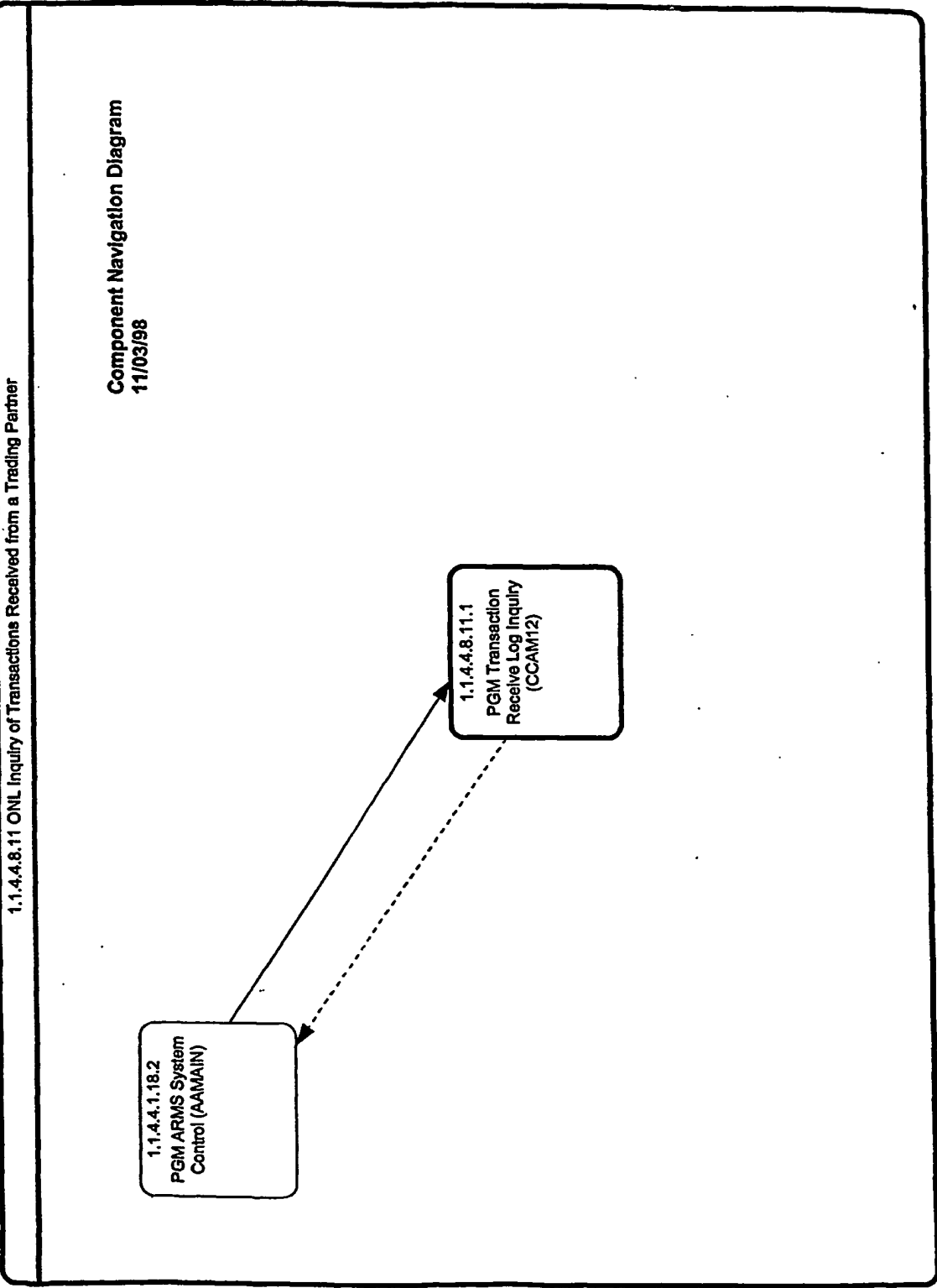




1.1.4.4.8 ONL Inquire on Transaction Log

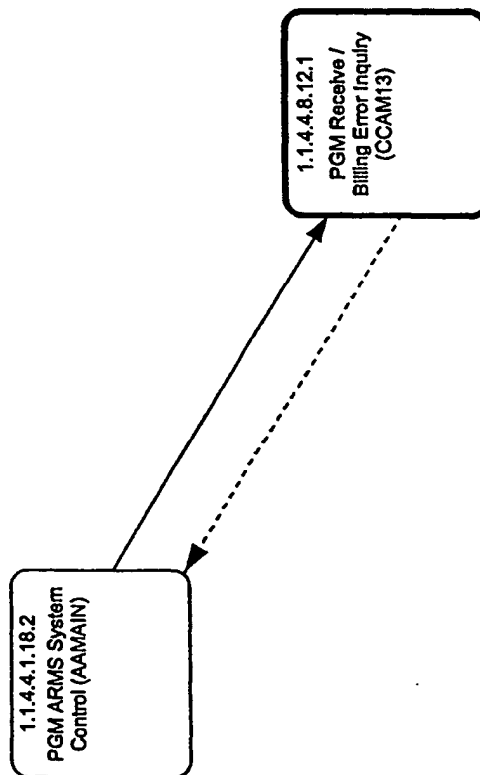
Screen Navigation Diagram  
01/27/99 GWE

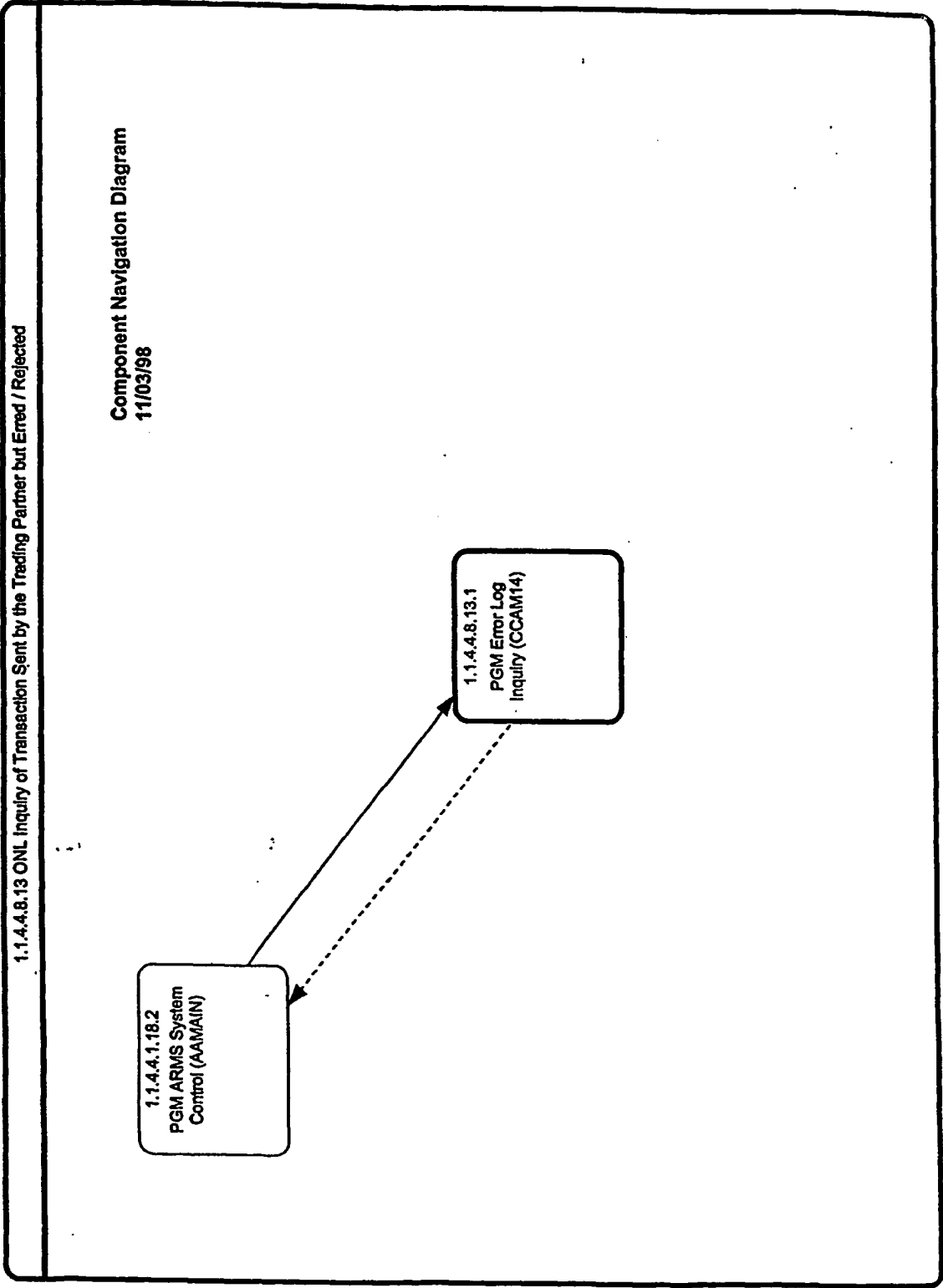


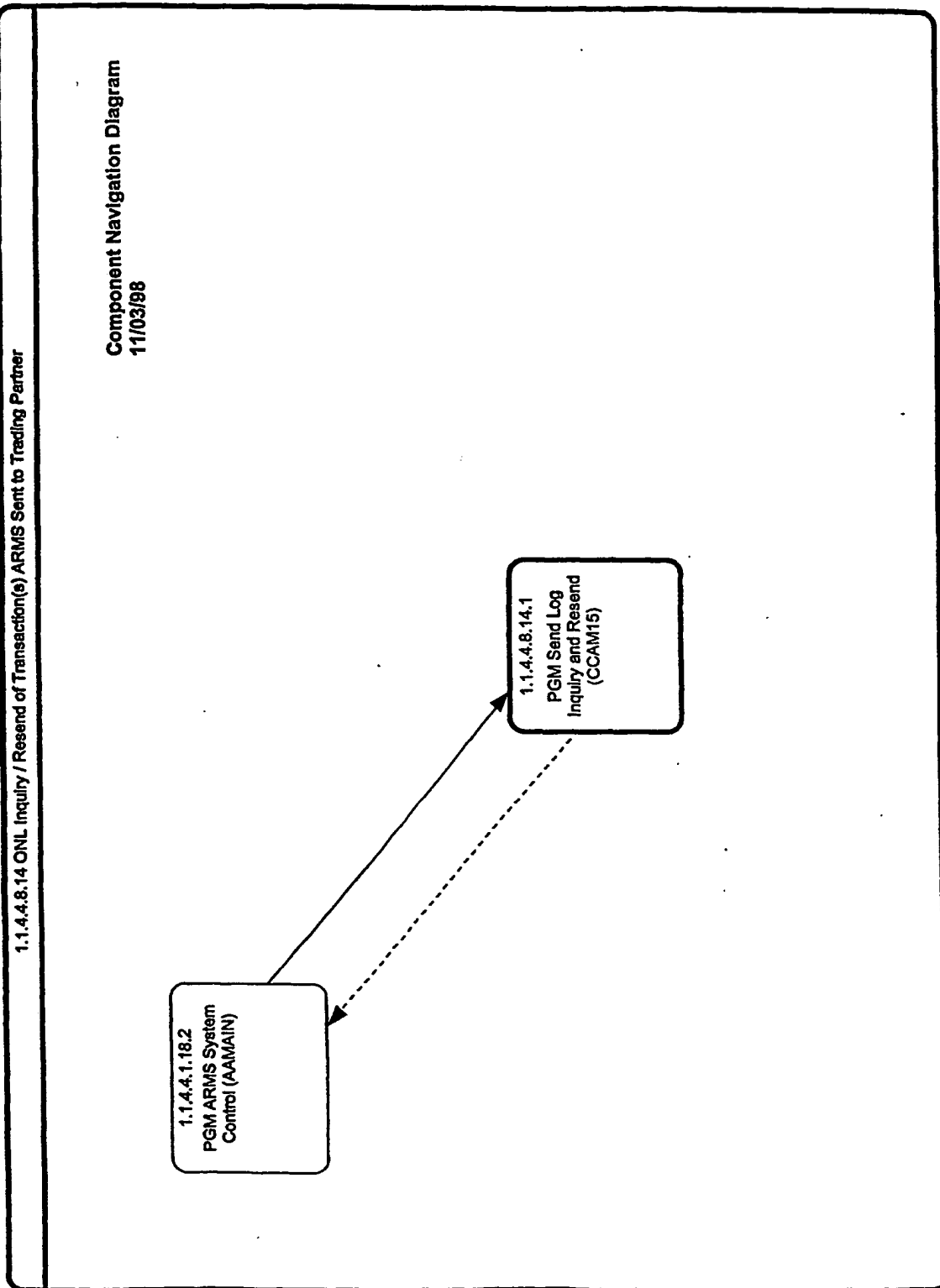


1.1.4.4.8.12 ONL Inquiry of Error Transactions Sent by Trading Partner

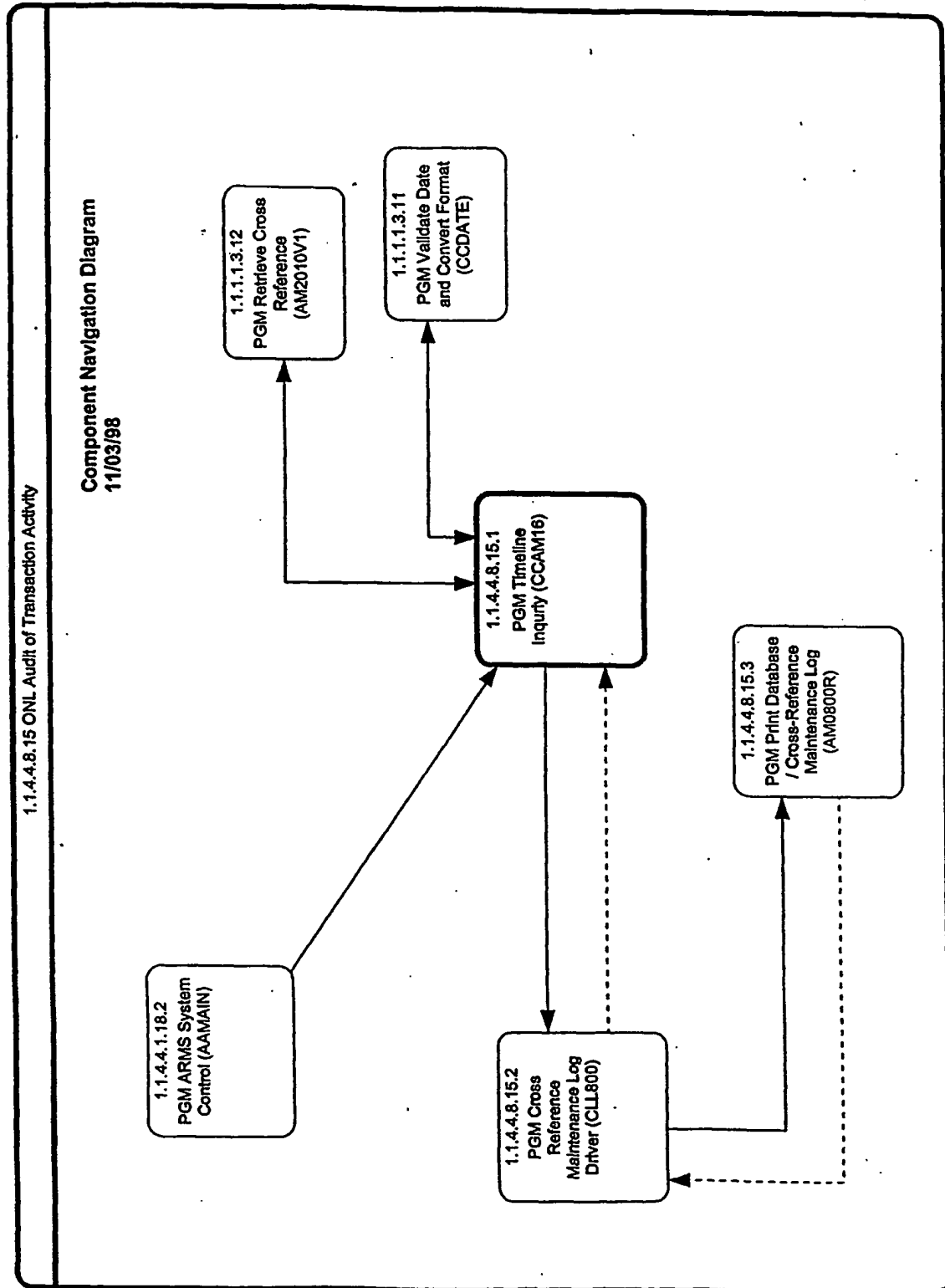
Component Navigation Diagram  
11/03/98

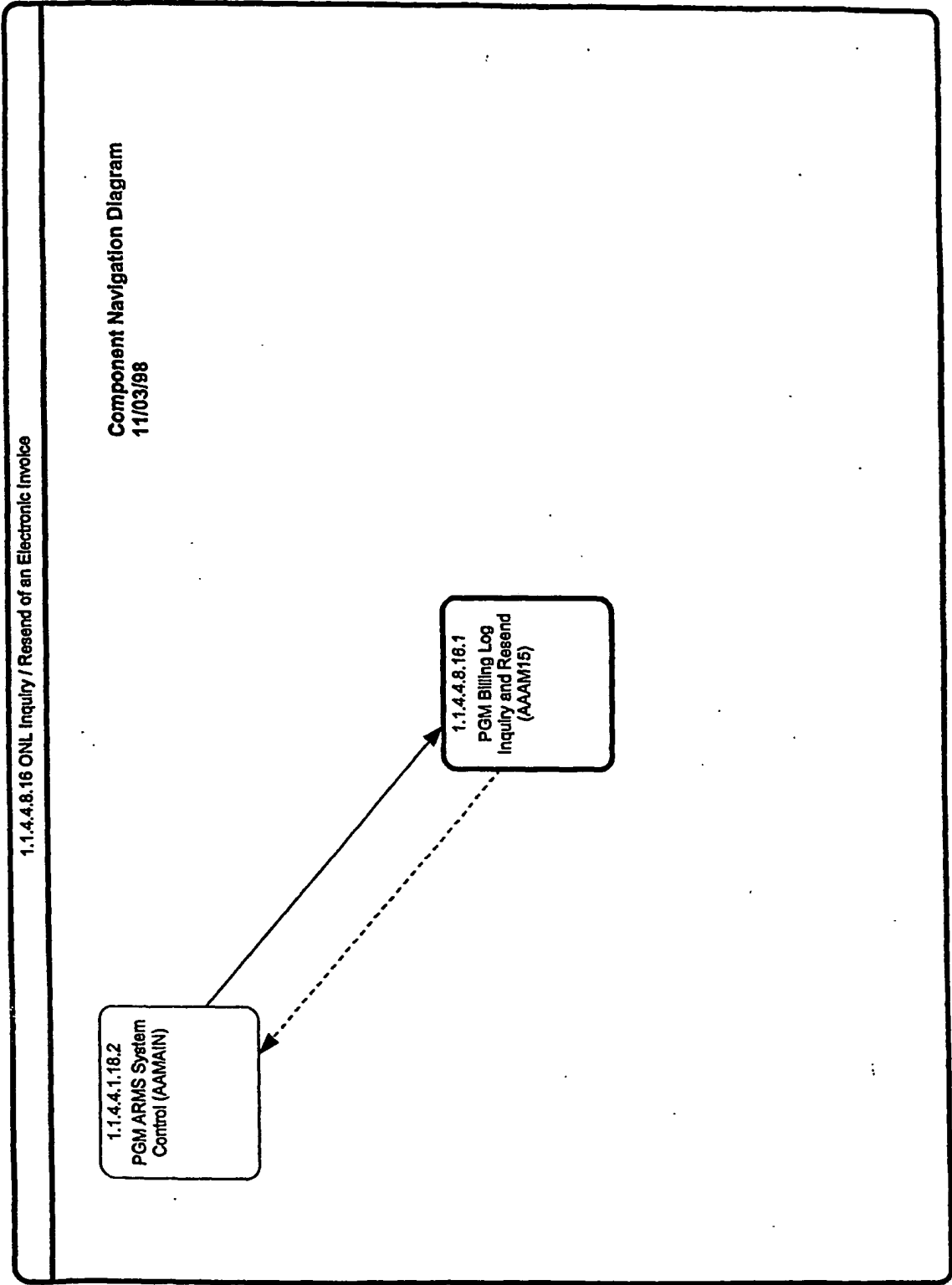


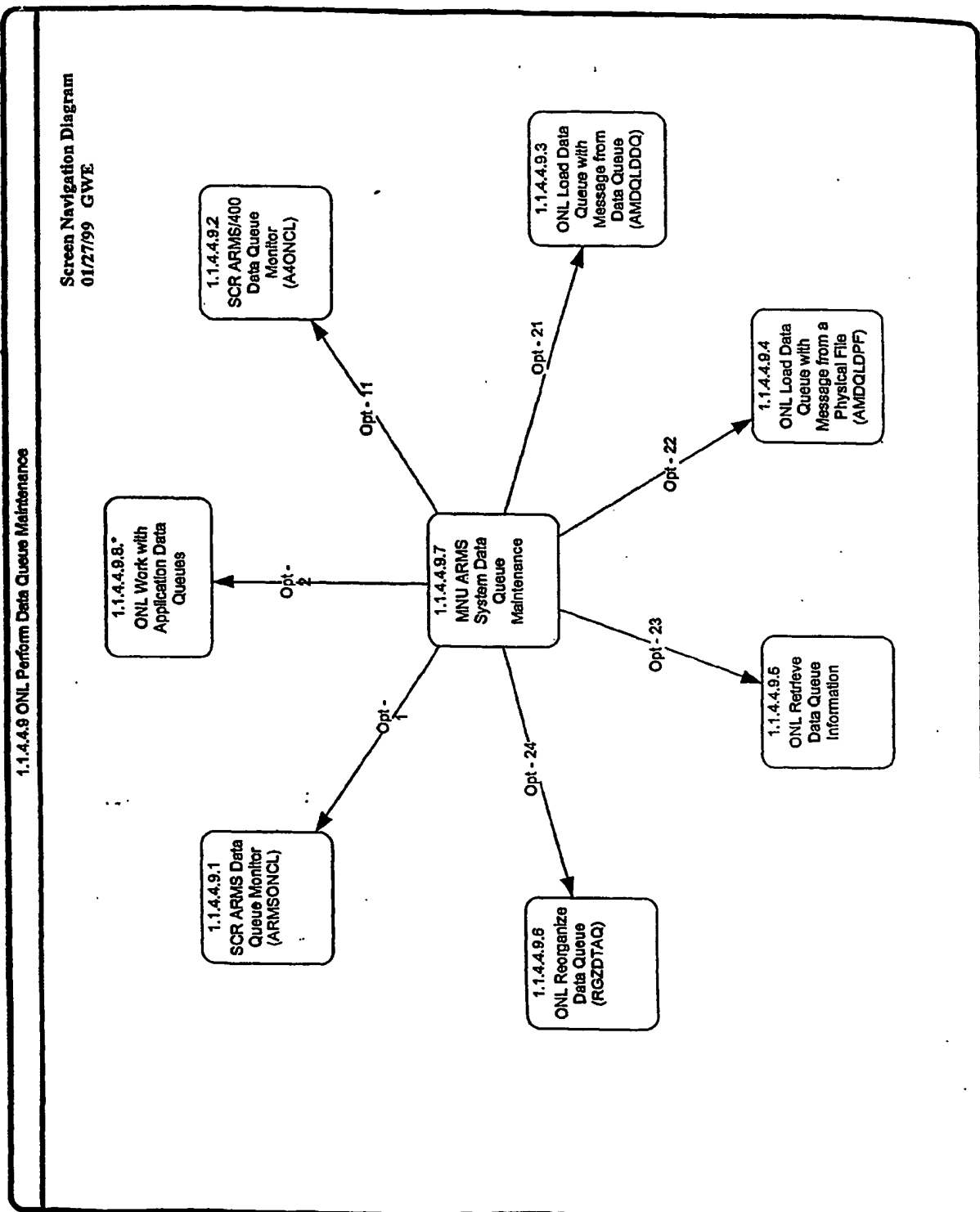






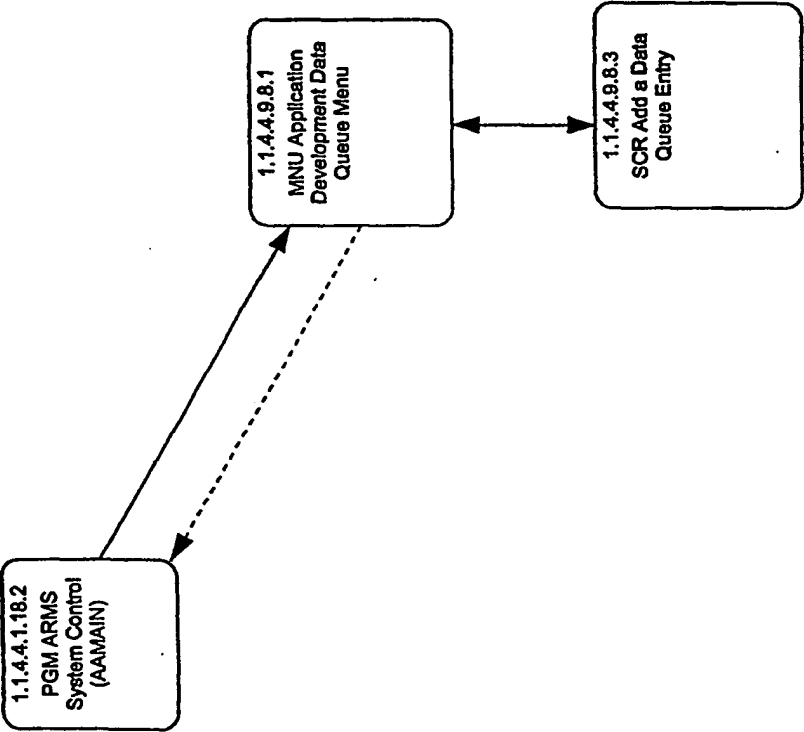


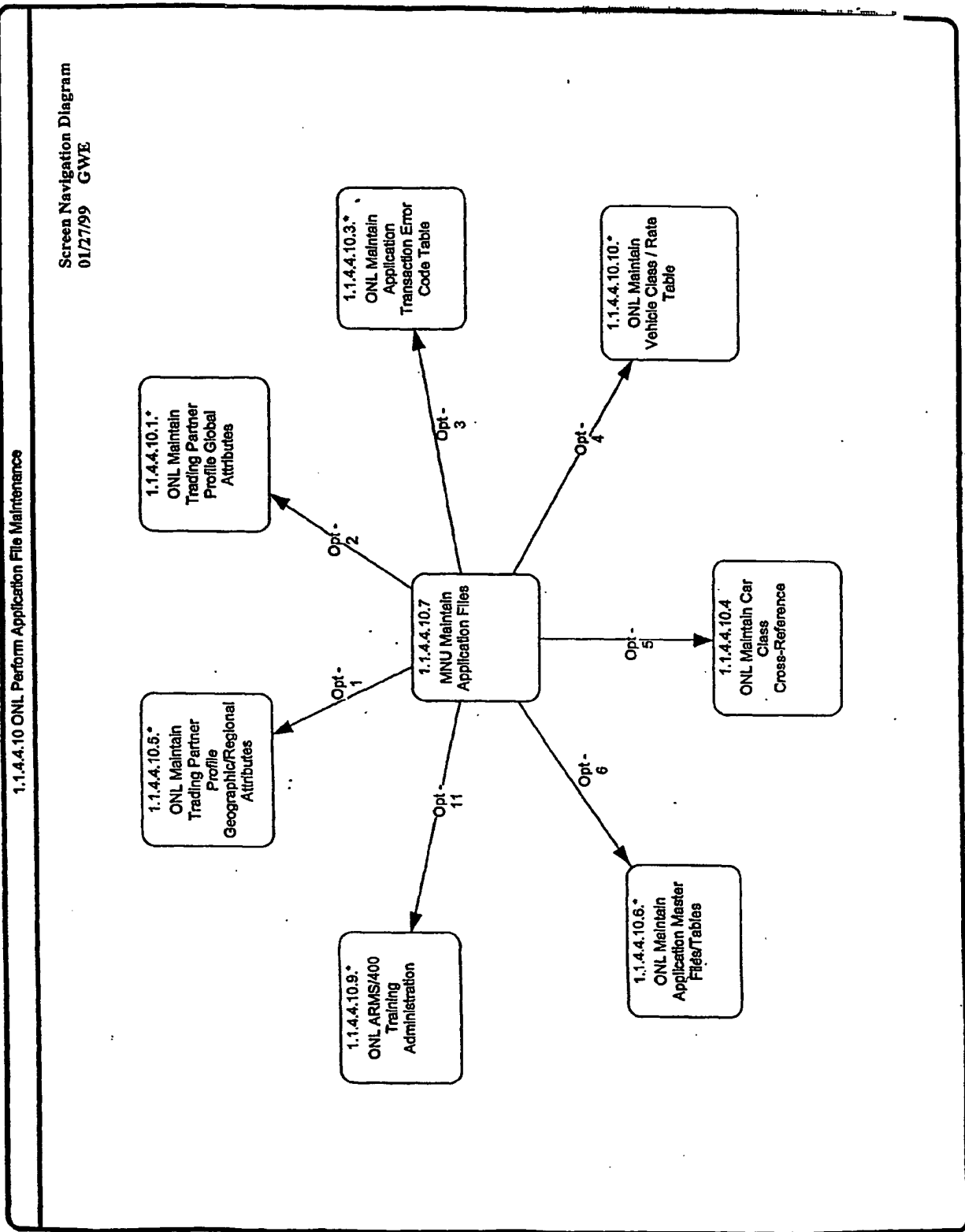


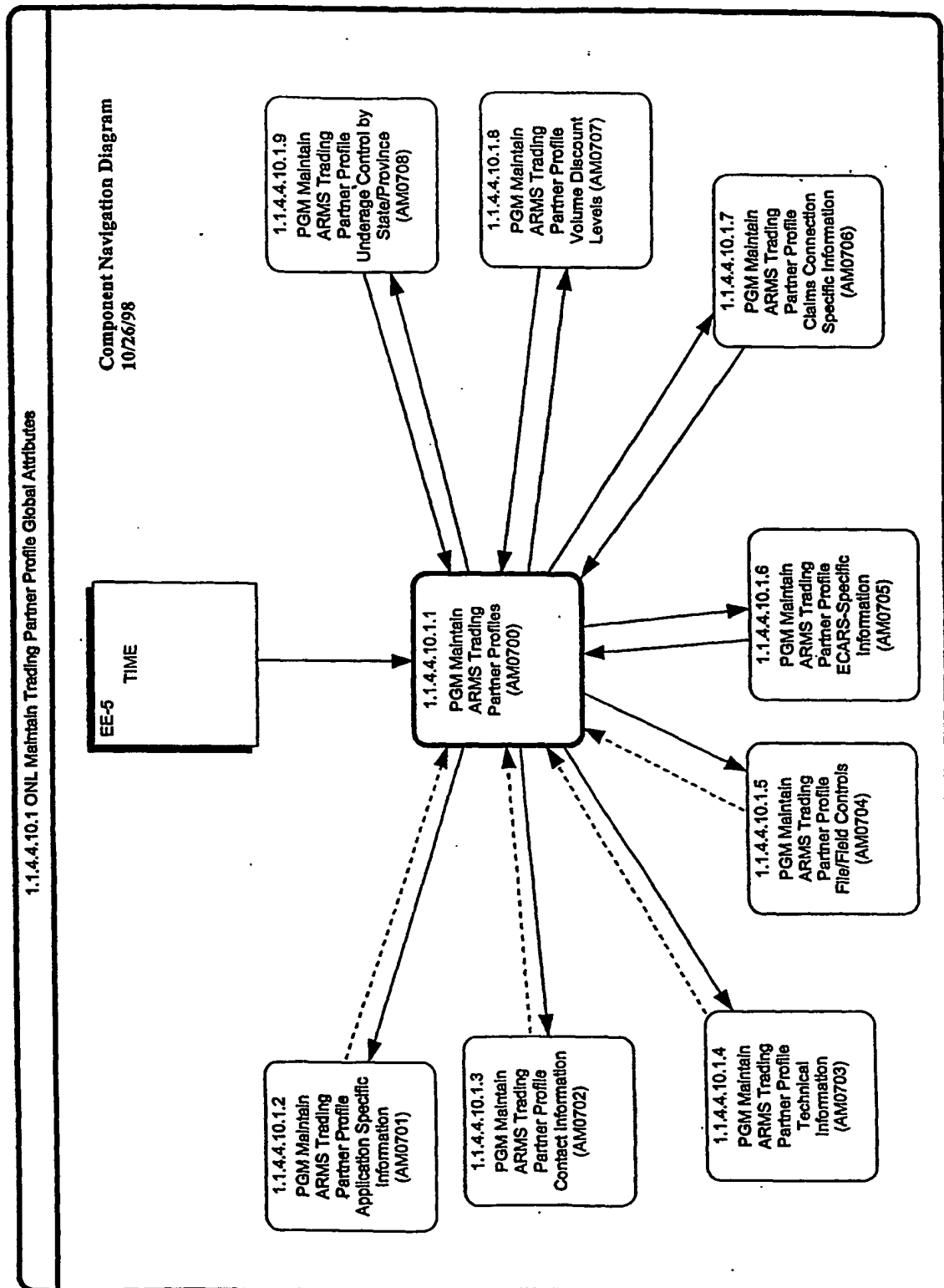


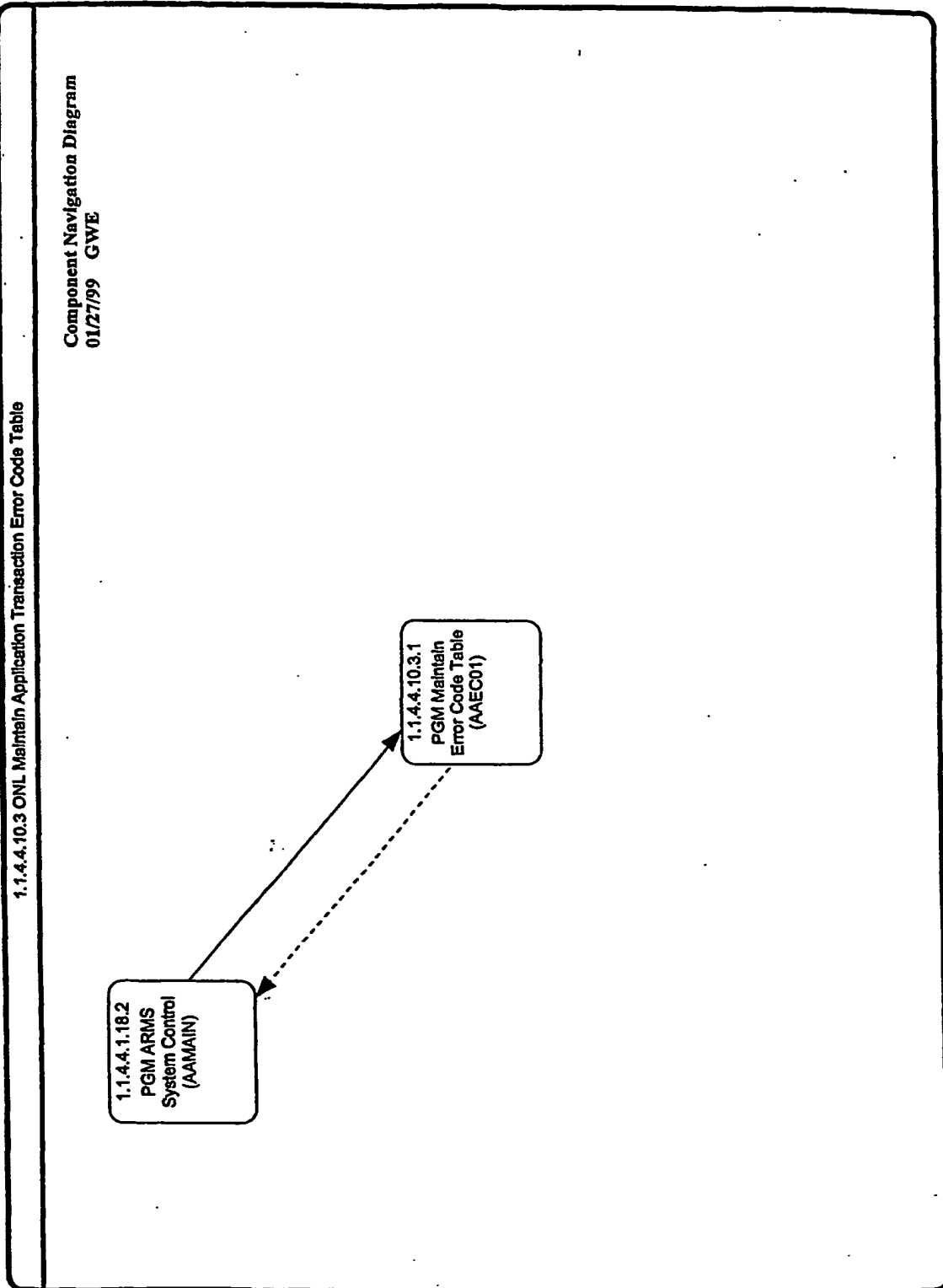
1.1.4.4.9.8 ONL Work with Application Data Queues

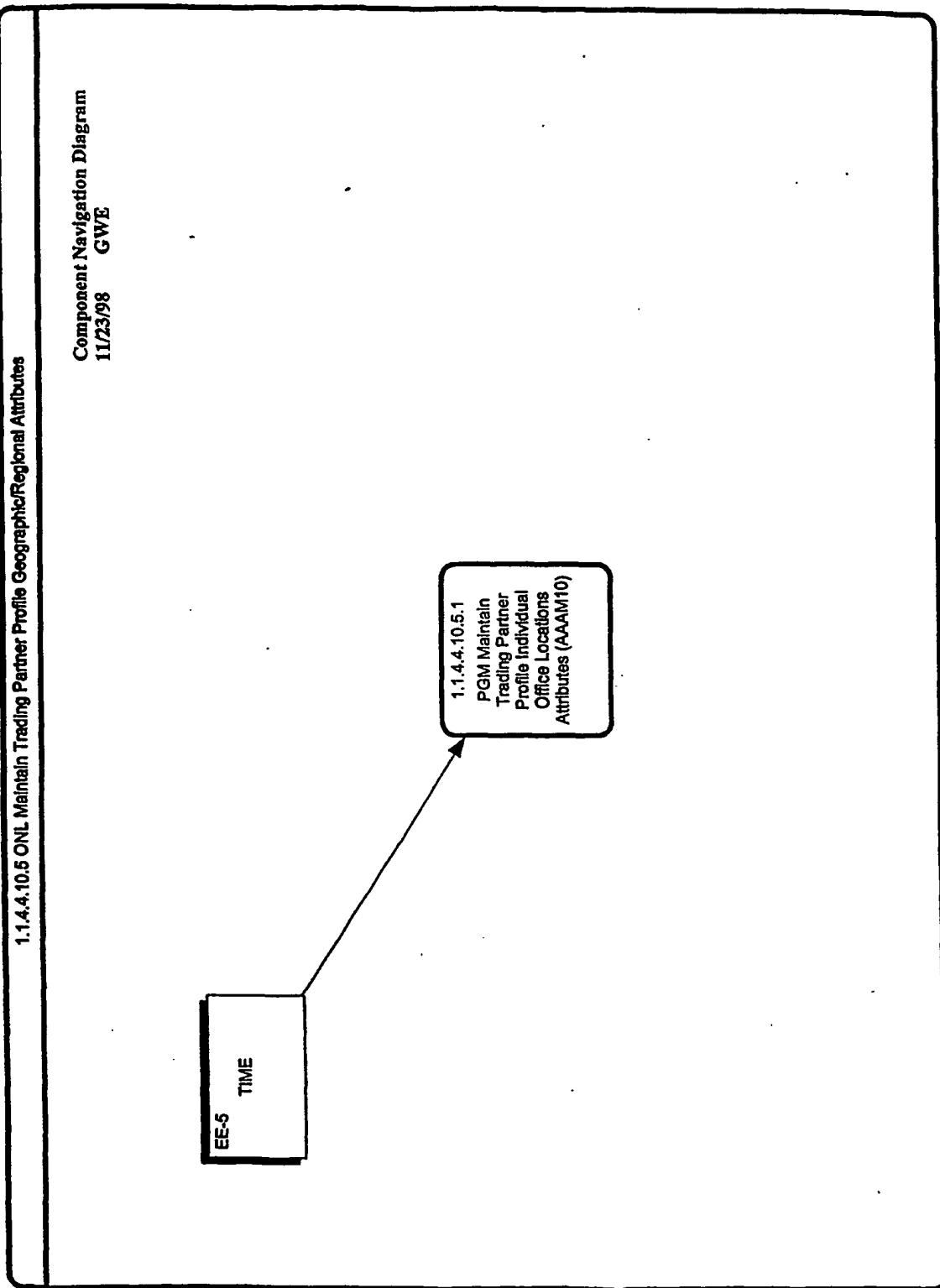
Component Navigation Diagram  
11/03/98







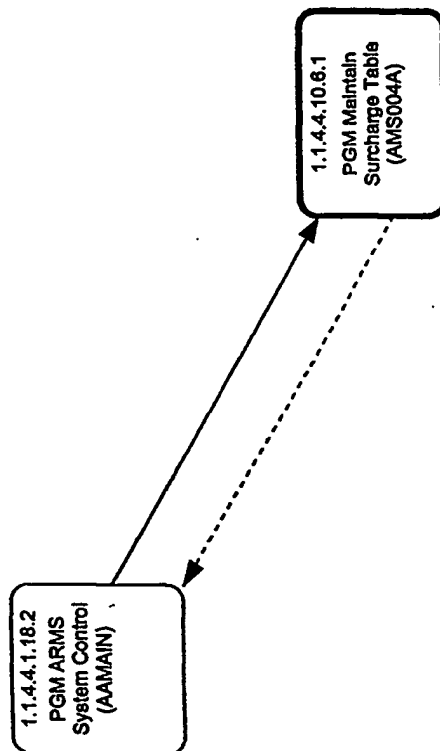






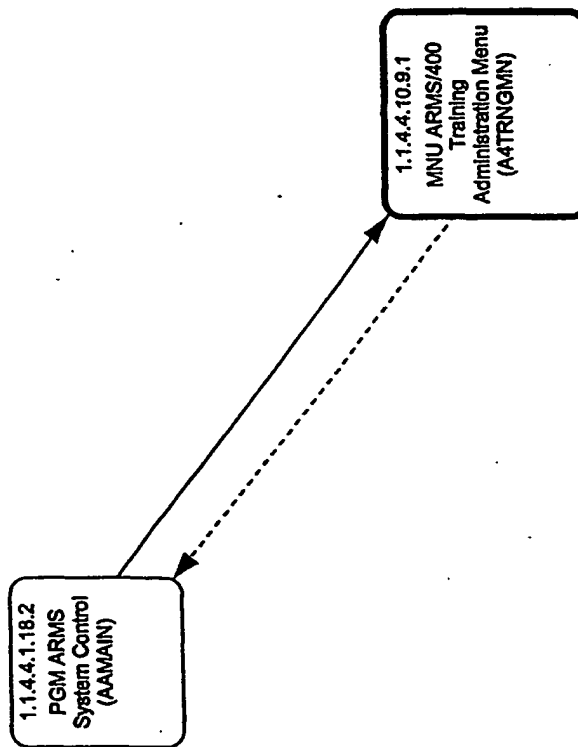
## 1.1.4.4.10.6 ONL Maintain Application Master Files/Tables

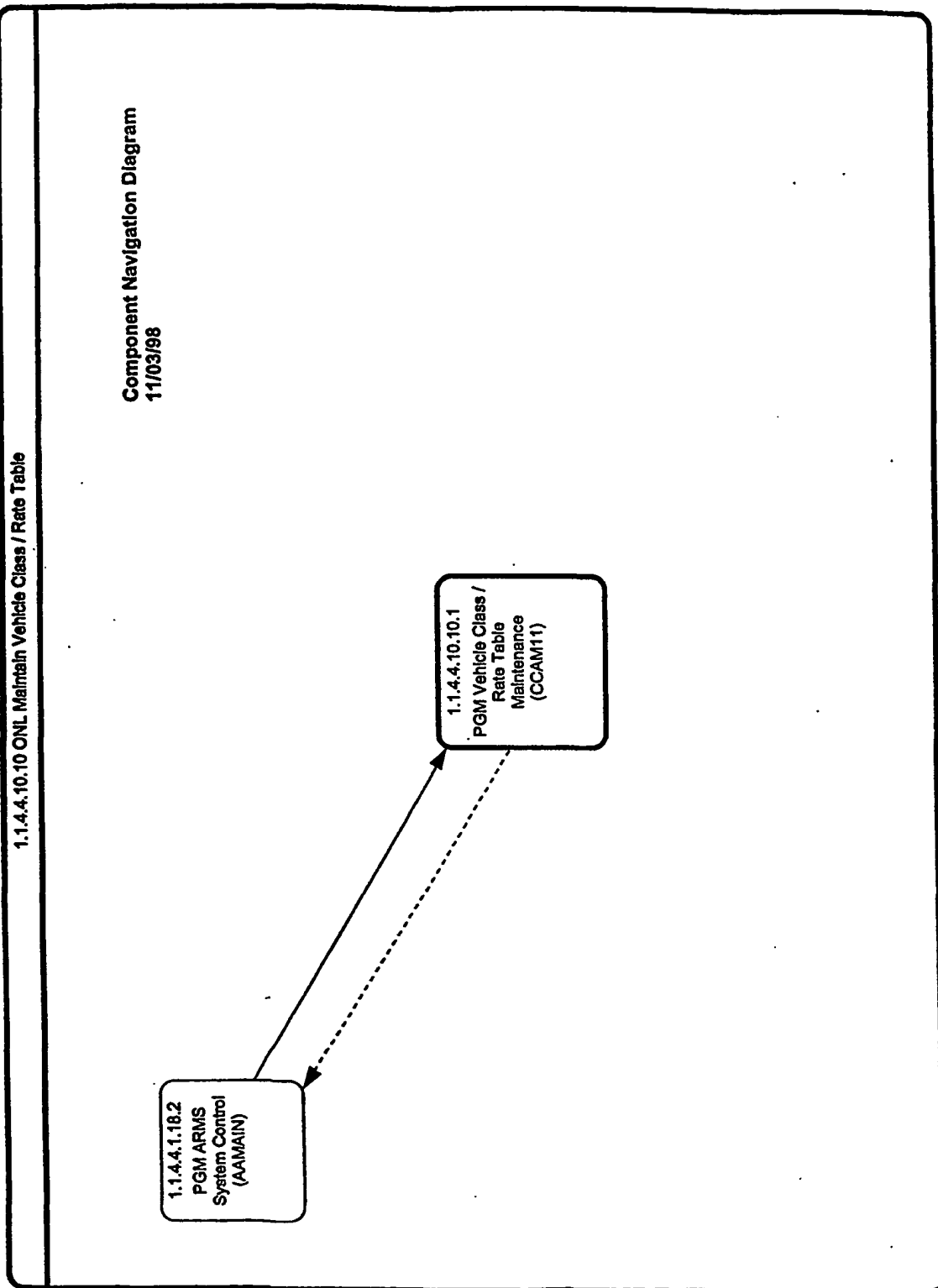
Component Navigation Diagram  
10/28/98 GWE



1.1.4.4.10.9 ONL ARMS/400 Training Administration

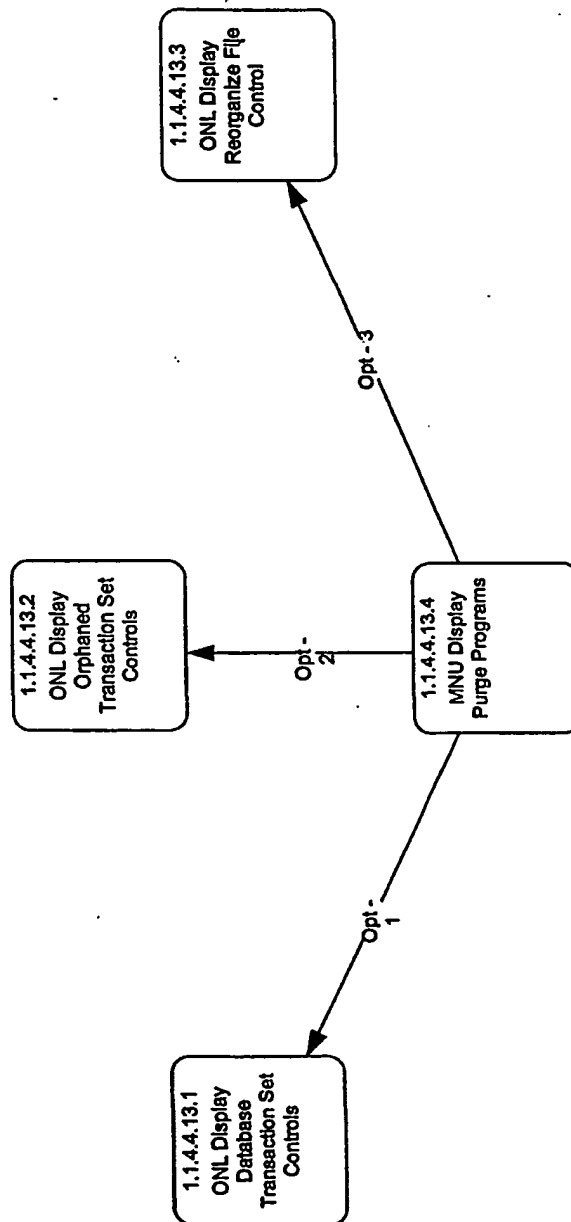
Component Navigation Diagram  
11/03/98





## 1.1.4.4.13 ONL Monitor Purge Activity

Screen Navigation Diagram  
01/27/99 GWE



**ARMS Process Report****EXHIBIT C****Process**

Hierarchical numeric ID: 1

Coded name:

Name: ARMS Context (CSD)

Comment: @Definition: Automated Rental Management System is the electronic data interchange (EDI) application and communications bridge between the Enterprise Rent-A-Car rental applications (ECARS-Enterprise Computer-Assisted Rental System and Claims Connection) and the application systems used by Rental Management Trading Partners (mostly insurance companies using ARMS/400, a VAN - Value-Added Network, or their own in-house application) and Direct Billing Trading Partners.

@Purpose: To accept EDI transmission envelopes from the trading partners, unpackage the transmissions, perform transaction data set edits, route to the appropriate distributed rental location's rental application system host computer platform and update the rental application database with the EDI transaction's information. To also, generate transaction data sets from the rental applications for sending to the trading partners.

@Notes: ARMS has a centralized data store for progressive transaction daily work files, received and sent transmission logs, error logs, billing logs, rental transaction database of information passed to date, trading partner transaction/rental transaction cross-reference, and trading partner profile files. ARMS has a distributed data store for transaction daily work files, error logs, trading partner transaction/rental transaction cross-reference and a subset of the duplicated images of the centralized trading partner profile files.

Also, the current transaction formats processed with some trading partners and within the majority of the ARMS software are using documented proprietary EDI transaction data set formats and transmission envelopes. However, some trading partners are exchanging information using ANSI (American National Standards Institute) X12 EDI standards for the 272 Rental Management transaction format that require translation to and from these proprietary EDI transaction data set formats, mapping, repackaging/unpackaging, and communication using an installed third-party software package, Any EDI integration software's "EDI Integrator".

**Process**

Hierarchical numeric ID: 1.1

Coded name:

Name: BA Automated Rental Management System

Comment: @Definition: This is a rental business area application system software set of processes that allows Automated Rental Management, rental reservations, rental location and rate maintenance notifications, invoice billing and electronic funds transfer (EFT) remittance advice transactions via EDI (Electronic Data Interchange) communications to be routed between trading partners and rental application and financial software systems within Enterprise Rent-A-Car.

**Process**

Hierarchical numeric ID: 1.1.1

Coded name:

### ARMS Process Report

#### AA Trading Partner Business Transactions

**at:** @Definition: This application area represents all EDI transmissions that sent from the EDI Rental Management Trading partner users to be routed to the tributed application system for:

creating rental reservations; perform rental payment authorization maintenance for a current rental contract or reservation; or, messages to the rental branch office staff. Also, will process received electronic remittance advice detail in response to electronic billing invoice for a closed rental contract that they had authorized payment to be routed to the Cash Receipts financial application system.

#### Process

Hierarchical numeric ID: 1.1.1.1

Coded name:

Name: BT Request for Authorization Management by Customer

**Comment:** @Definition: The set of automated activities that are used in the creation of reservations and maintenance of the rental payment/direct billing authorization from an EDI transmission received from the Rental Management Trading Partner.

@Purpose: To allow the Rental Management Trading Partner an electronic means of creating rental reservations and managing their rental payment/direct billing responsibility authorization information for a vehicle rental.

#### Process

Hierarchical numeric ID: 1.1.1.1.1

Coded name:

Name: AUT Receive Transmission

**Comment:** @Definition: This represents any of the external (not between the multiple Enterprise Rent-A-Car AS/400 computer systems) connection-specific communication receiver programs that receive communication transactions from an electronic data interchange (EDI) trading partner or their VAN - Value-Added Network service provider.

#### @Operational Method:

This program requests a program start from company; then continuously receives data from company until a record with \*DOWN is received indicating that this was the last transmission before the trading partner's sender program shutdown. For every non-shutdown transmission received, a data queue entry is sent to DQAM70V1 so that the received transmission is placed into the Received Transmission Log file (AMTRNLOG) as an audit trail and then the transmission is then output to the connection-specific received transmissions for either production or test purposes work file (AMxxxxxx), where 'xxxxx' is the PROFILE ID of the trading partner or his value-added network for production or testing, and 'n' is the version number - currently all are the value '1').

@Notes: Each direct connect trading partner or VAN (Value-Added Network), has a PRODUCTION PROFILE ID and a TEST PROFILE ID.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.1

Coded name: AMzzzRCV (zzz = Root of Van ID)

Name: PGM Receive Proprietary Transmission (AMzzzRCV)

**ARMS Process Report**

Comment: @Purpose: To receive proprietary EDI transmissions from any ARMS-connected VAN (Value-Added Network).

@Operational Method:

This program is evoked by a VAN's sender and then continuously awaits and receives data from the VAN until a record with \*DOWN is received.

The input is logged and put in the appropriate receiver file.

For any program exception error, the \*PSSR subroutine is executed to DUMP the current program and if the return code is equal to 8196 and the current time is outside of the 11:00pm - 3:00am time interval, call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Notes: For specific ICF operations, see the record formats in the ARMICF file that this program uses for input.

The "real-time" receiver job abnormally ends execution when the VAN's shutdowns their communication environment at 1:00am nightly, 12:00am Sunday. This program would normally end when receiving a \*DOWN request after having sent the VAN a \*STOP request command from the AMSTOP2R program executed nightly at 12:50am. Any abnormal halts or ending of the production communication receiver job during their scheduled operational time currently notifies the ARMS On-Call staff in order that they may manually intervene to notify any ARMS-connected VAN to attempt to restart the critical communication process job.

Process

Hierarchical numeric ID: 1.1.1.1.1.2

Coded name: CLL818

Name: PGM Set Environment to Receive Proprietary Batch Transmission (CLL818)

Comment: @Definition: This is the ARMS Trading Partner Company - to -ARMS receive ARMS-connected Value-Added Network batch communication driver program (written in CL/400 - Original Program Model).

@Purpose: This program's purpose is to set up the environment and call the necessary program to receive an electronic data interchange batch transmission envelope formatted in Enterprise Rent-A-Car's ARMS- proprietary EDI formats, over a dialup communication line, to IBM ARMS-connected Value-Added Network electronic communication network mailbox. This mailbox contains one or more transmissions intended to be received by ARMS, sent from the ARMS Trading Partner Company's Insurance Claims Management application system's associated batch sender communications programs on their IBM mainframe computer network system.

@Operational Method:

- This program is delay until 7:00am prior to any further processing.
- The next step is to vary on (activate) the ARMS-connected Value-Added Network communication line, EDI controller (includes all downline attached configuration elements) and the EDI integration software device if any of these communication elements are found to be not ready.
- Add EDI integration software library to the batch job's library list.
- Then for a production batch receive, call/execute program AM\*\*\*RCB with parameter '\*\*\*01' (ARMS Trading Partner Company production Profile ID) to Receive

### ARMS Process Report

Trading Partner Company Receive Batch file via ARMS-connected Value-Added Network with EDI integration software. This will write all records to ARMS Trading Partner Company Receive file and ARMS Receive Log file and send all records' transmission key value to the any ARMS Trading Partner Company Receiver Data Queue.

- The next step is to copy the records from the AM\*\*\*RCB ARMS Trading Partner Company Received Batch Transmissions file to an archive file, AM\*\*\*RCBSV and then clear the AM\*\*\*RCB file.

- Repeat the call /execute, copy and clear for testing using the parameter 'T\*\*\*T' (ARMS Trading Partner Company test Profile ID).

@Notes: Runs on the centralized ARMS application host computer system once per day, Monday through Saturday, after ARMS Startup's Restore Unprocessed Transaction batch job (L814) has completed, regardless if holiday. Scheduled via ROBOT/Scheduler REACT mode.

This program was created by the ETD- Emerging Technologies Department's (formerly Advanced Technologies Department) EDI project team, but is currently maintained primarily and the responsibility of the ARMS Application Development Department. This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice.

@Files: (CRUD)

- AM\*\*\*RCB (-R-D)
- AM\*\*\*RCBSV (C---)

#### Process

Hierarchical numeric ID: 1.1.1.1.1.5

Coded name: AM\*\*\*RCB

Name: PGM Receive ARMS-connected Value-Added Network Batch for specified ARMS Trading Partner Companies (AM\*\*\*RCB)

Comment: @Purpose: This program's purpose is to receive an electronic data interchange batch transmission envelope formatted in Enterprise Rent-A-Car's ARMS-proprietary EDI formats, over a dialup communication line, to IBM ARMS-connected Value-Added Network electronic communication network mailbox. This mailbox contains one or more transmissions intended to be received by ARMS, sent from the specified ARMS Trading Partner Companies Insurance Company's ARMS-interfacing application system's associated batch sender communications programs on their host computer network system.

@Operational Method: This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice. It will retrieve the EDI integration software Profile record for account 'ERAC' and user ID 'ERACARM' from the IEPROFL IBM EDI integration software User Profile File, load the externally-defined local data area with program execution entry parameter values, and call the IBM EDI integration software licensed program product EDI library's IEIFEXEC main program for IBM EDI integration software BASE. This program will then read the IEIFEXEC program's output AM\*\*\*SNB file records, that were "pulled" from specified ARMS Trading Partner Companies' send to ARMS mailbox on the IBM ARMS-connected Value-Added Network electronic communication network. Any production transmissions will be written to the AM\*\*\*011 specified ARMS Trading Partner Companies production received transmission output file and its key sent to the DQ\*\*\*011 specified ARMS Trading Partner Companies' Received Transmission Input Data Queue to its specific AM0010V1 program for transmission envelope validation. Any



### ARMS Process Rep rt

test transmissions will be written to the AM\*\*\*011 specified ARMS Trading Partner Companies production received transmission output file and its key sent to the DQT\*\*\*T specified ARMS Trading Partner Companies Received Transmission Input Data Queue to its specific AM0010V1 program for transmission envelope validation. For every input transmission record format received, each record format is written to the files AMTRNLOG (transactions received audit trail) and to file AMxxxxxx and sent to data queue DQxxxxxx (company 'xxxxxx' and version 'n' specific) for either production (\*\*\*\*01) or test (T\*\*\*T) purposes. For any program exception error, the \*PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Frequency: Once per day, Monday through Saturday after ARMS Startup's Restore Unprocessed Transaction batch job (L814) has completed, regardless if holiday.

@Notes: This program is called twice within a batch job's driver CL program, CLL818 ETD: ARMS Trading Partner Company Receive Batch program that runs on the centralized ARMS application host computer system. The first occurrence for this program execution with entry parameter to cause the program to perform only receive transmissions for the \*\*\*\*01 ARMS Trading Partner Company production processing. The second program execution with entry parameter T\*\*\*T to cause the program to perform only receive transmissions for the T\*\*\*T ARMS Trading Partner Company test processing. See documentation for CLL818 to determine when processed.

This program should be processing only when the pair of any ARMS-connected VAN communication sender program is active and the centralized ARMS application host computer system and its ARMS subsystem is active. Usually should be in operation from 3:00am to 1:00am, Monday through Tuesday and on Sunday from 7:30am to 11:00pm.

This program was created by the ETD- Emerging Technologies Department's (formerly Advanced Technologies Department) EDI project team, but is currently maintained primarily and the responsibility of the ARMS Application Development Department. This program is used to receive a batch transmission envelope that might contain the following Enterprise ARMS proprietary EDI transaction sets: Authorization maintenance (AT-Add and AT-Change), ER-Error, and PM-Payment Advice.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.6  
 Coded name: EDMRLUA  
 Name: PGM Receive LU 6.2 Transmission (EDMRLUA)  
 Comment: @Definition: This is the ARMS Receiver LU 6.2 communication program.

@Purpose: This program's purpose is to receive an electronic data interchange transmission envelopes formatted in CIECA drafts for ANSI X.12 standard EDI formats, over a leased communication line, with an ARMS Trading Partner Insurance Company's automobile claims application system's associated sender communications program on their host computer network system.

@Operational Method: 1. Capture job attributes and set up library list, debug options, logging, traces, etc. 2. Retrieve parameters from data area with the same name as the job = same as the device description. (This allows control of various options without requiring that the initiator of the APPC ALLOCATE send parameters to control them.) 3. Set up communications line trace and/or ICF trace as specified. If the current user does not have sufficient authority, proceed without traces. If any other trace error, proceed without traces. 4. Execute either the EDISYS/STRCTLSCRIB (API: Start ctl script to call ARMS-connected Value-Added Network)

### ARMS Process Report

or the EDISYS/STRRCVTRNB (API: Start ctl script; rcv/unwrap/transmission) command using appropriate parameters.

@Frequency: Heavy utilization between 6:00am - 3:00pm (all times are Central time zone), Monday through Friday, especially at 6:00am receiving batch burst of scheduled invoice payment advice (PM) transaction set transmissions. Additional heavy utilization from 10:00pm - 10:10pm receiving batch burst of scheduled invoice payment remittance advice (RM) transaction set transmission batches. Moderate utilization between 7:00am - 9:00am and 3:00pm - 7:00am Monday through Friday. Light utilization on all other active times, including weekends and holidays.

@Notes: This program was created and is maintained primarily by the Emerging Technologies Department (ETD - formerly Advanced Technologies Department) EDI project team. This program is used to receive all real-time business-event transaction data transmission sets for production or test information interchange. This is running as a prestart job because this will enable the programs involved in this job to be pre-started so that the programs and their resources remain open and active since each trading partner evokes the communication session each time they send a transmission envelope.

@Process Frequency: This program is used within two never-ending program prestart batch jobs on the centralized ARMS application host computer system. One job is Lxxxxx1R (where xxxxxx is the full ARMS Company Profile ID) that executes EDxxxxRC1 (where xxxxxx is the full ARMS Company "root" Customer ID) - ETD: EC/EDI: ARMS X.12 Trading Partner Receiver Communication for Production. That program is started by remote via an APPC MC\_ALLOCATE. In order for APPC (Advanced Program-to-Program Call) to trigger it properly, there should be a communications entry in the any EDI integration software subsystem which associates the device or remote location to JOBID EXCMNJOBID and user EDIOWN. This program can be executed from either a prestart job command within the AM\*\*\*COMP (where \*\*\* is the Profile ID root value) program or by an ARMS subsystem "prestart" job that executes ED\*\*\*RC1 program, that calls/executes this EDMRLUA program with a single entry parameter for the ARMS Profile ID value to cause the program to perform "receive" production transaction processing. The other job, ED\*\*\*RCT, executes this program with entry parameter 'T\*\*\*X' will cause the program to perform only "receive" test transaction processing. The production receiver job starts execution at usually on Sunday at 8:00am and stays active 24 hours per day through Saturday at 1:00am in order to do RARMS mirrored systems role swap when this "real-time" receiver job is normally ended. Any exception errors or abnormal ending that occur during the execution of the production communication receiver job on the centralized ARMS application host computer system during their scheduled operational time currently causes the notification of the ETD and ARMS primary and backup On-Call staff pagers. This is done so that they may manually intervene to attempt to restart the critical communication process job.

@Data Areas: (CRUD)

- EX || Company Id (-R--) (in EDI integration software system library)
- EXDBG (-R--) (in any EDI integration software system library)

#### Process

Hierarchical Numeric Ed. : 1.1.1.1.1.7

Coded name: EDMRMQA

Name: PGM Receive Alternate Protocol Transmission (EDMRMQA)

Comment: @Definition: This is the ARMS Receiver for alternate protocol communication program.

@Purpose: This program's purpose is to receive X12 standard EDI formats, over a leased communication line, with ARMS Trading Partner Insurance Company's

<b>ARMS Process Report</b>
----------------------------

automobile claims application system's associated sender communications program on their host computer network system.

@Operational Method: 1. Capture job attributes and set up library list, debug options, logging, traces, etc. 2. Retrieve parameters from data area with the same name as the job = same as the device description. (This allows control of various options without requiring that the initiator of the APPC ALLOCATE send parameters to control them.) 3. Set up communications line trace and/or ICF trace as specified. If the current user does not have sufficient authority, proceed without traces. If any other trace error, proceed without traces. 4. Execute either the EDISYS/STRCTLSCRB (API: Start control script to call ARMS-connected Value-Added Network) or the EDISYS/STRRCVTRNB (API: Start control script; receive/unwrap transmission) command using appropriate parameters.

@Notes: This program is maintained by the ETD and is used to receive all real-time transaction sets for production or test environment. As of July 6, 1998, it has not been put into regular scheduled daily use because a specified ARMS EKDI Trading Partner Company's application changes to exchange with ARMS has not yet been implemented.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.8  
 Coded name: EDMRUWA  
 Name: PGM Unwrap Envelope (EDMRUWA)  
 Comment: @Purpose: To unwrap received X12 EDI Transmission Envelope.

#### @Operational Method:

- Receive and unwrap an electronic data interchange transmission envelope formatted in ANSI X12 standard EDI formats.

- For any program exception error, the ATDPSSR and AMPSSR programs are called to send a pager messages to the ETD and ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

@Notes: This program is maintained by the ETD. This program is used as a never-ending program batch job on the centralized ARMS application host computer system.

Currently used to receive transmissions over a leased communication line, with ARMS Trading Partner Insurance Company's automobile claims application system's associated sender communications program on their mainframe computer network system, based in Boston, Massachusetts.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.9  
 Coded name: EDLRMIA  
 Name: PGM Receive Mapper Interface (EDLRMIA)  
 Comment: @Purpose: To drive and control the translation of the X12 EDI formats in the incoming transmission to proprietary formats.

#### @Operational Method:

- Declare programs variables

### ARMS Process Report

- Change the Any EDI integration software communication script name to the passed input parameter field value for COMPANY ID
- Allocate the receive interface files
- Wait for EDI transaction envelope
- IF transactions have been received, execute the any EDI integration software's Start Translate Connections Command to search for Log of Connection records which are eligible for incoming translation.
- IF translation error encountered, Notify ETD and ARMS on-call staff and end job.
- ELSE, Call the ARMS EDI Receive Interface program to write application records.
- IF program is running in debug mode, Copy receive interface files to their save files.
- Clear the receive interface files
- ELSE, (transactions have not been received)
- Deallocate receive interface files.
- Send shutdown data queue entry to the receive unwrap input data queue.

@Notes: This program was created and is maintained primarily by the Emerging Technologies Department (ETD - formerly Advanced Technologies Department) EDI project team. This uses the any EDI integration software's mapping/translation utility third-party software package.

This program is submitted as a called never-ending program batch job with the job name 'LxxxxxLRM' (where 'xxxxx' is the Trading Partner Profile ID) and specifies using the any EDI integration software system library's EDISFTWJOBDF job description on the centralized ARMS application host computer system by the 'L\*\*\*011R' receiver job with the following input parameters:

5 Alphameric COMPANY ID  
 1 Alphameric DEBUG FLAG (possible values: "Y" (Yes) or "N" (No))

(12/17/98)

This object which is used in the receive mapper job has been changed to delay until midnight at shutdown before sending a shutdown data queue entry to the receive unwrapper job. This change was necessary because transactions received between 11:45 p.m. and midnight would not get functional acknowledgements sent back to ARMS Trading Partner Insurance Company and would not show up on our daily reporting because the receive unwrapper job wasn't active until midnight.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.10  
 Coded name: EDZRCUA  
 Name: PGM Get Mapped Data (EDZRCUA)  
 Comment: @Purpose: This program's purpose is to check if any ANSI X12 EDI data has been received. This program also checks for shutdown.  
 @Operational Method:

This program is a called program that accepts three (3) parameter fields: Company Profile ID (input-5 alpha), Shutdown Flag (output-1 alpha) and Transactions Received Flag (output-1 alpha).

When executed, initialize the SCRIPT ID with the passed COMPANY PROFILE ID value concatenated with "R" (Receive).

Default load the TRANSACTIONS RECEIVED FLAG to "N" (No).

ARMS Process Report
---------------------

If the passed COMPANY PROFILE ID is a production profile (ends in "1"), then change the receive mapping data queue name to 'DQ' + the first three characters of the company ID + 'RM1'. Else, change the receive mapping data queue name to 'DQ' + the second through fourth characters of the company ID + 'RMT'. Then continue the following processing until data has been received and mapped or a shutdown data queue entry has been received.

Receive a data queue entry from the receive mapping data queue on infinite wait. Once a data queue entry is received, change the receive data queue operation from infinite wait to 15 seconds.

Attempt to retrieve records from the Any EDI integration software Connection Log logical file by SCRIPT ID with Unwrapped Receive Status. If there are records, change the output transactions flag to 'Y' (Yes).

If the data entry received is '\*DOWN', then load the SHUTDOWN FLAG parameter with 'Y' (Yes) and end the program.

For any program exception error, the \*PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated. Also, write the transaction the program was processing to the ERRORS file.

@Files: (CRUD)

- EXLLCPFR (-R--) \* Any EDI integration software Connection Log by Script Name with Unwrapped Receive Status

(in EXTSYSF Any EDI

integration software EDI Integrator Files library.)

- ERRORS (C---) \* Errors Log (User-Controlled Open)

@Notes: This program is maintained by ETD. This program is used within two never-ending program batch jobs on the centralized ARMS application host computer system. One job executes this program with entry parameters to cause the program to perform only "receive" transaction processing, the other job executes this program with entry parameters to cause the program to perform only "send" transaction processing. The receiver jobs usually start execution at Monday through Saturday at 5:20am and Sunday at 1:30pm. The "real-time" receiver jobs abnormally end execution when the EDI trading partners shutdown their communication environment at 11:00pm. These programs will normally end when receiving a \*STOP request from the AMSTOP2R program executed nightly at 12:50am. Any abnormal halts or ending of these jobs during their scheduled operational time currently notifies the ARMS On-Call staff in order that they may manually intervene to attempt to restart the critical communication process job.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.12

Coded name: AM\*\*\*COM

Name: PGM Receive/Send Proprietary Transmission for ARMS Trading Partner Company (AM\*\*\*COM)

Comment: @Definition: This is the ARMS-to-ARMS Trading Partner Company (and vice-versa) SNUF (SNA Upline Facility) (IBM) 3270 terminal emulation device API transaction-based (triggered) communication program.

@Purpose: To receive or send an proprietary EDI formats, over leased ARMS-connected Value-Added Network lines, with the ARMS Trading Partner Company

### ARMS Process Report

Insurance Company's Insurance Claims Management application system's associated sender and receiver communications programs on their host computer system.

#### @Operational Method:

This program requests a program start from company, then continuously awaits and receives data from ARMS Trading Partner Company's computer system until a record with \*DOWN is received.

When input is received, each record is written to the files AMTRNLOG (transactions received audit trail) and to file AMxxxxxx (Company Profile ID 'xxxxxx' and Version 'n' specific) for either production or test purposes.

For any program exception error, the \*PSSR subroutine is executed to DUMP the current program and call AMPSSR program to send a pager message to the ARMS primary and backup On-Call pagers so that manual intervention may be initiated.

#### @Files: (CRUD)

```
- ARMSPR1      (-R--)
- ARMSPR3      (-R--)
- AM***011     (C--) (specified as ANDUMPRD)
- AMT***TT     (C--) (specified as ANDUMTST)
- AMSEND       (-R--)
- AMTRNLOG     (C---)
- AMSNDLOG     (C---)
```

@Notes: This program is maintained by ETD.

#### Process

Hierarchical numeric ID: 1.1.1.1.1.13

Coded name: EDxxxxRC1

Name: PGM Prestart Job by Company for LU6.2 (EDxxxxRC1)

Comment: @Definition: These are any programs that cause the jobs to be "prestarted" to define the system resources used in the communication prior to being sent any communications transmissions by a trading partner's remote system. A prestart job is a batch job that starts running before a program on a remote system sends a program start request. Prestart jobs are different from other jobs because they use prestart job entries to determine which program, class, and storage pool to use when they are started. Within a prestart job entry, you must specify attributes that the subsystem uses to create and manage a pool of prestart jobs.

@Purpose: The purpose of this program is to receive EDI formatted transactions from Rental Management Trading Partner and write them to the inbound control and detail files.

#### @Notes:

- This program is activated by ARMS Trading Partner Insurance Company, using an ALLOCATE verb. The corresponding ACCEPT verb makes the connection through ARMS-connected Value-Added Network using LU62.

- This program will received a complete EDI formatted 272 transaction set as one continuous long record. This long record is parsed out into 80 byte records and written to the detail file INBDTA. One corresponding control record is written to INBCTL for each complete transaction set received.

### ARMS Process Report

- A last batch number is retrieved from dtaara EDIA001 and updated. This batch number is the key for the detail and control files.

- A Comm Profile record is also updated with a batch sequence number.

- A Confirm Request is received, and this program replies with a Confirmed response.

- A Deallocate is received and this pgm writes the control file record and then returns to the ACCEPT verb and waits for another ALLOCATE from ARMS Trading Partner Insurance Company. This sequence is then repeated.

- This program is a PRE-START program and is started with the sbs. It remains active even when not connected to ARMS Trading Partner Insurance Company. It will set forever on the ACCEPT verb, waiting for ARMS Trading Partner Insurance Company to ALLOC.

- This program does NOT go to EOJ when it receives the DEALLOCATE from ARMS Trading Partner Insurance Company, but re-establishes the connection by repeating the Accept, Receive, Confirmed, and Deallocate sequence.

- EOJ is indicated when a STOP record is received or a CPI-C return code indicates an error. It then must be manually restarted by the STRPJ command (start pre-start job).

#### Process

Hierarchical numeric ID: 1.1.1.1.1.14

Coded name: AMZRMIA

Name: PGM Receive X12 Interface (AMZRMIA)

Comment: @Purpose: To process X12 transmission set(s) within the receive interfaces files by mapping the elements from the transmission set data record to the associated ARMS proprietary record formats.

#### @Operational Method:

- Process the Electronic Commerce Customer Preference file (EDPREF3).

- IF transaction type equals Rental Management for Vehicle Replacement Rentals (X12 272)

- Determine application transaction type(AK, AT, CN, EX, or ER)
- Map associated X.12 element to corresponding proprietary record

format

- Write data format records to transmission received file (AMTRNLOG)

- IF transaction type equals Rental Management Remittance Advice for Vehicle Replacement Rentals (X12 820)

- Map associated X.12 element to corresponding proprietary record

format

- Write data format records to transmission received file

(AMTRNLOG)

- IF transaction type equals Rental Management for Vehicle Replacement Rentals (X12 824)

- Determine application transaction type (RM-rejected, ER)
- Map associated X.12 element to corresponding proprietary record

format

ARMS Process Report
---------------------

- Write data format records to transmission received file
---

(AMTRNLOG)

## @Notes:

This program was created and is primarily maintained by the ETD - Emerging Technologies Department's EDI team.

ETD is currently working with the following X12 transaction sets that affect rental:

272 Rental Management - 2 maps, 1 for ARMS Trading Partner Company and a CIECA one which will be used by ARMS Trading Partner Company

824 Consolidate Payment - 2 maps, 1 ANSI X12 for ARMS Trading Partner Insurance Company and a CIECA X12 one which also will be used by a different ARMS Trading Partner Insurance Company.

820 Payment - CIECA map used by ARMS Trading Partner Company (who is a non-ARMS customer)

811 Consolidated Invoice - CIECA map for consolidated invoicing that ARMS Trading Partner Insurance Company and TVA are using.

832 Pricing Catalogue - map for ARMS Trading Partner Insurance Company which we are using to exchange branch rate and availability information.

The implementation guide for this was given to us by ARMS Trading Partner Company and that is what we used to get this going.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as any Enterprise Fleet Services customer.

@Files:	(CRUD)
R272HDR1	(-R--)
R272NMH1	(-R--)
R272DTL1	(-R--)
R272LQ1	(-R--)
R272COM1	(-R--)
R272ER1	(-R--)
R997AK1	(-R--)
AMEDI011	(C--)
AMTRNLOG	(C--)
AMPGMERR	(C--)
R820HDR1	(-R--)
R820DTL1	(-R--)
R820ADX1	(-R--)
R820NTE1	(-R--)
AMXREFL1	(-R--)
EDPREF3	(- R--)
AMAUTD	(-R--)
ARMSPR1	(-R--)
AM805P01	(CR--)
AM806P01	(CR--)
AMSNDLOG	(-R--)
EDPREF3L1	(-R--)
R824HDR1	(-R--)
R824DTL1	(-R--)



ARMS Process Report
---------------------

R824TED1	(-R--)
R824NAM1	(-R--)
R824NTE1	(-R--)
AM804P02	(-R--)
AMXREF	(-R--)
AMCLSTB2	(-R--)
AM801P02	(-R--)

## @Embedded Data/Constants:

FILE(AMEDIO11) TOFILE(AM ) Used for override to appropriate trading partner/VAN production/test receive file

## @Improvement Opportunity

- Use program status data structure to retrieve program name.

## Process

Hierarchical numeric ID: 1.1.1.1.1.16

Coded name: AM\_001

Name: PGM Retrieve ARMS Trading Partner Company Password (AM\_001)

Comment: @Purpose: To retrieve the connection password for a ARMS Trading Partner Company from the data area so as to restrict the authority to these data areas to a certain specified number of people.

## @Operational Method:

Retrieve data area AMSFPWPR if this is a production request.  
Retrieve data area AMSFPWTS if this is a test request.  
Return the password.

## Process

Hierarchical numeric ID: 1.1.1.1.1.18

Coded name:

Name: DTQ Input for Receive Unwrapper (DQxxxRU1)

Comment:...

## Process

Hierarchical numeric ID: 1.1.1.1.1.19

Coded name:

Name: DTQ Input for Receive Mapper Interface (DQxxxRM1)

Comment:

## Process

Hierarchical numeric ID: 1.1.1.1.1.20

Coded name: EDMRPIA

Name: PGM Receive X12 Transmission Batch (EDMRPIA)

Comment: @Purpose: To receive and unwrap the X12 transmission from a direct billing trading partner.

## @Operational Method:

- Receive the actual transmission into a temporary buffer

**ARMS Process Report**

- CALL EDZ001A to retrieve direct billing trading partner's communication connection information. If the return code is '1', this means that there is no transmission to process otherwise use the returned communication NETWORK, PORT and SCRIPT information to retrieve the transaction data and put it into the application interface files.

- CALL AMZRMIA to convert retrieved data in interface files to proprietary formats for further processing.

@Notes: Currently, this program is called by a scheduled job (LTVA011RUM) that runs at 2:00 pm to receive a batch of transmissions from Tennessee Valley Authority (TVA)

**Process**

Hierarchical numeric ID: 1.1.1.1.1.21

Coded name: EDZ001A

Name: PGM Retrieve Communication Parameters (EDZ001A)

Comment: @Purpose: To retrieve the direct billing trading partner's communication information.

@Operational Method:

- Read file EDPREF1 to retrieve the direct billing trading partner's PORT, NETWORK and SCRIPT information.

- If no record is found a return code of 1 indicating nothing can be processed.

**Process**

Hierarchical numeric ID: 1.1.1.1.2

Coded name:

Name: AUT Validate AND Unpackage (CommRcvr - AM20/AM21)

Comment: @Definition: The automatic process of separating from the received and validated EDI transmission envelope the individual transaction data sets so each base transaction is processed further individually.

**Process**

Hierarchical numeric ID: 1.1.1.1.2.2

Coded name: AM0010V1

Name: PGM Validate Transmission (AM0010V1)

Comment: @Definition: The ARMS never-ending batch program to validate the completeness of a transmission envelope's contents.

@Operational Method:

At startup do the following steps:

1. Read, increment by 1 and update the DASDV1 data area value to indicate that another AM0010V1 NEP job has started.

2. Construct the derived input data queue name from "DQ" concatenated with the TRADING PARTNER/VAN PROFILE ID input parameter value concatenate with the constant "1".

<b>ARMS Process Report</b>
----------------------------

3. Override the derived ARMS Input file (AMINPUT) to the ARMS Input TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1. Then open the input file.

4. Override the derived ARMS Hold file (AMHOLD) to the ARMS Hold TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1. Then open the hold file.

After startup, continue to receive the next data queue entry from the derived input data queue until program shutdown.

For a shutdown data queue entry received:

1. Read the DASDV1 data area value, decrement by 1, update it.
2. Send a shutdown data queue entry to the Unpackaging Input Data Queue (DQAMTRN1).
3. End the program/job.

For each non-shutdown data queue entry received:

1. Use its value to read by key the next TRADING PARTNER/VAN RECEIVER INPUT FILE record.
2. IF not end-of-file, check the retrieved record's FORMAT ID:
  - A. IF the record's FORMAT ID is part of the packaging (Transmission, Group, or Set Start/End), then load the record to that appropriate proprietary record format external data structure.
    - i. IF the SETE01 format has been read, check the derived TOTAL DETAIL FORMATS READ COUNTER is less than 1.
      - a. IF true, reject the transmission envelope
    - B. IF the record's FORMAT ID is NOT part of the packaging (i.e. the transaction's information), check if at least one Transmission, Group and Set Start record format was read.
      - i. IF false, then reject transmission with appropriate error code.
      - ii. IF true, then write it to the ARMS Hold TRADING PARTNER/VAN RECEIVER INPUT FILE Version Number 1 and increment by one the derived TOTAL DETAIL FORMATS READ COUNTER.

- Validate if received transmission envelope is complete. If not valid, generate an Error Transaction (ER) data set record formats to the Transactions to be Packaged for Sending (AMPACK) file and its key to the associated input (DQAMPKG) data queue, and write an associated record in the AMERRLOG file.

- Generate a return Communication Acknowledgement (AK) transaction set for the valid transmission envelope received.

**@Notes:**

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) when executed on the system known as "RARMS" host computer system platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input. Normally, this can indicate that its corresponding receiver or receiver's mapper-interface never-ending program batch jobs that had been previously active that day have processed a shutdown request.

It is submitted from the CLL810 ARMS Start-Up program with the following input parameter fields:

5 Alpha      TRADING PARTNER/VAN PROFILE ID

<b>ARMS Process Report</b>
----------------------------

8 Alpha	TRADING PARTNER/VAN RECEIVER INPUT FILE
---------	---

@Files: (CRUD)

AMHOLD (CR-D) (HLxxxxx1)

ARMSPR1 (-R--)

AMERRLOG (C---)

AMxxxxx1 (-R--) (Where 'xxxxx' is the connect-specific Trading Partner or VAN production or test Profile ID)

AMTRN (C---)

AMPACK (C---)

DASDV1 (-RU-) Data Area: number of AM0010V1 program/jobs running, used by AM0020V1 to check when last AM0010V1 program active has shutdown.

@Embedded Data/Constants:

'0123456789' for testing for valid numbers.

OVRDBF FILE(AMINPUT) TOFILE(AMCMPID1) to override to the appropriate AMxxxxx1 file.

OVRDBF FILE(AMHOLD) TOFILE(HLCMPID1) to override to the appropriate HLxxxxx1 file.

CLRPFM FILE(\*LIBL/HLxxxxx1) to execute a clear physical file member on the appropriate HLxxxxx1 file.

### Process

Hierarchical numeric ID: 1.1.1.1.2.3

Coded name: AM0020V1

Name: PGM Unpackage Transaction from Transmission (AM0020V1)

Comment: @Purpose: To unpackage received proprietary transmission and remove the transmission control formats (also referred to as transmission envelope) TSMS01, TSME01, SETS01, SETR01, GRPS01 and GRPE01.

@Operational Method:

1. Receive the next data queue entry for this program's input data queue (DQAMTRN1).

2. For every shutdown data queue entry, retrieve, decrement by 1, and update data area DASDV1 until new value is zero. When it is zero, send a shutdown data queue entry to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and end the program. For further details about shutdown, refer to support activity "BAT Shutdown ARMS Environment".

3. For every non-shutdown data queue entry received:

A. Read all of the associated records (by the key value of the data queue entry) from the received packaged transmissions log file (AMTRN).

B. For each set start (SETS01) record format:

i. Attempt to use its PROFILE ID and its CUSTOMER TRANSACTION ID to retrieve an existing associated ARMS Cross-Reference File record. IF found, hold its VENDOR TRANSACTION ID and load its MACHINE ID, SOURCE ID, RENTAL LOCATION ID, RESERVATION ID, and TICKET ID values, along with the Transmission Group Start record format's GROUP TYPE CODE and the Transmission Group's Set Start record format's SET ACTION TYPE CODE values and this PROGRAM ID value and a ROUTING CODE of 'I', to their associated fields in the internal format APPD01.

<b>ARMS Process Report</b>
----------------------------

11. Load the format, APPD01, and all associated subsequent transaction data set record formats' derived key value with the current record's key, concatenated with the Transmission Group Start record format's GROUP CONTROL ID and GROUP TYPE CODE, and the Transmission Group's Set Start record format's CUSTOMER TRANSACTION ID, and,

IF the ARMS Cross-Reference File record was retrieved successfully, its VENDOR TRANSACTION ID value.

C. For each record format read that is not a Transmission, Group, or Set Start/End record format,

If the group transaction type is not 'ER', write the record format with the derived key value to the output file, Transaction Sets to be Edited Daily Transaction file (AMSET)

else

If the group type in error as specified on the ERRD01 record format is not 'IN', write the record format with the derived key value to the output file, Received Errors Log file (AMRCVERR)

else

Write the record format with the derived key value to the output file, Received Billing Errors Log file (AMBILERR).

D. For each Transmission Group Set End (SETB01) record format, send the current transaction data set's derived concatenated key value to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and finish reading the input file until the next Transmission Start (TSMS01), Group Start (GRPS01), or Set Start (SETS01) record format is read for the current read input key value.

**@Notes:**

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) with a single input parameter, VERSION, with the constant value of '1'. This never-ending batch program ends normally when a set of shutdown data queue entries are received as input that indicate that each AM0010V1 never-ending program batch jobs that was previously active that day have processed a shutdown request.

**@Files: (CRUD)**

AMTRN	(-R--)
AMSET	(C---)
ARMSPR3	(-R--)
AMAUTD	(-R--)
AMRCVERR	(C---)
AMBILERR	(C---)

**@Embedded Data/Constants:**

'\*DOWN00000000000000SD' is the key value used as output when sending a shutdown data queue entry to DQAM25V1.

'\*DOWN' and 'SD' are the literal constants used to check the data queue entry value loaded into the ARMSKEY external data structure's data elements for COMPANY PROFILE ID and GROUP TYPE CODE to determine if a shutdown was received.

'\*LIBL' is used as the literal constant to pass as an input parameter field value for the DATA QUEUE LIBRARY NAME for the calls to the 'QRCVDTAQ' and 'QSNDDTAQ' programs..

All of the possible ARMS Cross-Reference File Record's STATUS CODE values are used as literal constants in checking certain codes for conditional logic.

### ARMS Process Report

**@Improvement Opportunities:**

'\*DOWN00000000000000SD' could be used by many never-ending programs and the Send Data Queue Entry programs referenced from the Work with Data Queues menu. Therefore, place this value in an application system-wide data area and reference this external object.

Replace the literal constant '\*LIBL' with this value in an application system-wide data area and reference this external object.

**Process**

Hierarchical numeric ID: 1.1.1.1.2.4

Coded name: AM0021V1

Name: PGM Unpackage Transaction from Transmission (AM0021V1-RM Only)

Comment: @Definition: This is a never-ending batch program to unpackage any remittance advice functional groups from a proprietary EDI transmission envelope into individual internal transaction data sets with one or more record formats.

**@Purpose:**

This program will edit and unpackage individual remittance detail transaction data sets within Remittance Advice ('RM') functional groups within received transmissions' proprietary EDI envelopes. This program also matches and edits each transaction set's Profile ID and Customer Transaction ID to an associated existing ARMS Trading Partner Profile File and an ARMS Rental Transaction Cross-Reference File record. This is used to validate if the retrieved files' records' attribute indication settings and the Cross-Reference Status Code value should allow such action, and outputs each validated individual internal transaction data set with an corresponding internal proprietary (APPD01) header record format and a unique associated record key value to ARMS Transaction Sets to be Validated (AMSET) file and its associated data queue (DQAM25V1). The output transaction set will be processed as input by the Validate Transaction Set Record Formats (AM0025V1) program

**@Operational Method:**

1. Receive the next data queue entry for this program's input data queue (DQAM21V1).
2. For every shutdown data queue entry, send a shutdown data queue entry to the Edit Transaction Group/Set Record Formats data queue (DQAM25V1) and end the program.
3. For every non-shutdown data queue entry received:
  - A. Using the data queue entry's Trading Partner Profile ID value, retrieve the associated ARMS Trading Partner Profile Application Specific Information File record.
  - B. Read each of the associated records (by their key value of the received data queue entry) from the ARMS Validated Received Proprietary EDI Transmission Envelopes to be Unpackaged file (AMTRN).
  - C. For each Transaction Set Start (SETS01) record format processed:
    1. Load the AMTRN file record's key value (consisting of the TRADING PARTNER PROFILE ID and the TRANSMISSION CONTROL SEQUENCE ID NUMBER) to the AMSET file key and append to it the following:
      - a. the Functional Group Start (GRPS01) record format's GROUP CONTROL SEQUENCE NUMBER and the functional GROUP TYPE CODE.
      - b. the Transaction Set Start (SETS01) record format's CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID.

<b>ARMS Process Report</b>
----------------------------

ii. Load the Internal Header Record Format (APPD01) fields with the appropriate data.

D. For each remittance detail (RMTD01) record format:

i. Validate its PAYMENT CODE. IF invalid, reject with ERROR  
CODE='51'.  
- Edit to ensure there is a non-blank EFT ID NUMBER on all paid remittances.  
- Validate that any Rejected Payment Remittance Advice Detail Record Format does not have a value in the EFT ID NUMBER field.

ii. Use the retrieved AMTRN file record's key value's PROFILE ID and CUSTOMER TRANSACTION ID to attempt to retrieve an existing associated ARMS Cross-Reference File record. IF found, then validate that the Cross-Reference File record is the proper status. IF the read remittance detail's CUSTOMER TRANSACTION ID value does not match its retrieved ARMS Cross-Reference file record same field value, reject with ERROR CODE='82'. IF the status is valid, hold its VENDOR TRANSACTION ID and load its MACHINE ID, SOURCE ID, RENTAL LOCATION ID, RESERVATION ID, and TICKET ID values, along with the Transmission Group Start record format's GROUP TYPE CODE and the Transmission Group's Set Start record format's SET ACTION TYPE CODE values and this PROGRAM ID value and a ROUTING CODE of 'I', to their associated fields in the internal unpackaged transaction data set (APPD01) record format.

iii. Load the internal unpackaged transaction data set (APPD01) and all associated subsequent transaction data set record formats' derived key value with the current record's key, concatenated with the Transmission Group Start record format's GROUP CONTROL ID and GROUP TYPE CODE, and the Transmission Group's Set Start record format's CUSTOMER TRANSACTION ID, and,

IF the ARMS Cross-Reference File record was retrieved successfully, its VENDOR TRANSACTION ID value.

iv. Based on the trading partner profile, edit the remittance amount against the invoiced amount, and error the remittance detail record if the amount values are not equal. This comparison will not occur for RM reject or illegal transactions. The value of the profile attribute that indicates whether a trading partner sends remittances will be 'X' for those customers that can short pay and so the totals need not match (e.g. TVA).

v. IF an error is detected, call error program AM0090V1, which will create an ERRD01 format.

vi. For each validated RMTD01 record format read, write the record format with the derived key value to the output file of Transaction Sets to be Edited Daily Transaction file (AMSET) and send its key as a data queue entry to DQAM25V1.

E. For a Remittance Advice Total record format (RMTT01) read, IF its TOTAL AMOUNT or the TOTAL TICKETS does not balance with all of the non-rejected Remittance Advice Detail record format records' PAYMENT AMOUNT sum or a derived record count, error the Remittance Total record format, but continue to process all the detail records which have not errored. Rejected Payment Transactions may exist within Paid Transactions batches. \*Note, a remittance total format is not required when only one remittance advise is received and it's status is 'R'ected.

F. For any record formats read other than RMTD01 and RMTT01 and the Transmission, Group and Set Start/End record formats, create an APPD01 format especially for it. Then write out that invalid format and save the AMSET key to write to DQAM25V1 when the SETE01 record format is read.

ARMS Process Report
---------------------

**@Notes:**

This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform. This program is currently submitted by the ARMS Start-Up Job (CLL810) with a single input parameter, VERSION, with the constant value of "1". This never-ending batch program ends normally when a set of shutdown data queue entries are received as input that indicate that each AM0010V1 never-ending program batch jobs that was previously active that day have processed a shutdown request.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer).

**@Files:** (CRUD)

AMTRN	(-R--)
AMSET	(C---)
AMIEBT	(-R--)
AMIEBH	(-R--)
ARMSPR1	(-R--)

**@Embedded Data/Constants:**

'0123456789' Named constant to check for numeric fields.

Valid values for BILLING INVOICE STATUS in the RMTD01 record format:

'P' = Payment (Can be in full or a partial payment).  
 'R' = Rejected (must have additional ERROR CODES provided)  
 'X' = Illegal (must have additional ERROR CODES provided)

All of the following are the possible ARMS Cross-Reference File Record's STATUS CODE values allowable are used as literal constants in checking certain codes for conditional logic to allow remittances to be further processed. However, 'P' is only valid when the ARMS Trading Partner Profile (ARMSPR1) flag is set accordingly.

'PABQ'  
 'B' = Rental Contract Electronically Billed at least once  
 'A' = Passed Electronic Billing Audit  
 'P' = Paid in full  
 'Q' = Rejected for Payment by Trading Partner

'\*DOWN' and 'SD' (ShutDown) are the literal constants used to check the data queue entry value loaded into the ARMSKEY external data structure's data elements for COMPANY PROFILE ID and GROUP TYPE CODE to determine if a shutdown was received.

'AM0021V1' is used as a literal constant to pass as an input parameter field value to the calls of 'AMPSSR' and 'AM0090V1' programs.

'\*LIBL' is used as the literal constant to pass as an input parameter field value for the DATA QUEUE LIBRARY NAME for the calls to the 'QRCVDTAQ' and 'QSNDDTAQ' programs..

**@Improvement Opportunities:**



ARMS Process Report
---------------------

Replace the literal constant 'AM0021V1' with the Program Status Data Structure's data element for PROGRAM ID.

Replace the literal constant '\*LIBL' with this value in an application system-wide data area and reference this external object.

## Process

Hierarchical numeric ID: 1.1.1.1.2.5

Coded name: DQAM21V1

Name: DTQ Input for program AM0021V1 (DQAM21V1)

Comment: @Definition: DQAM21V1 is the data queue used to provide input to PGM AM0021V1 which processes remittance advises

## Process

Hierarchical numeric ID: 1.1.1.1.2.6

Coded name: DQAMTRN1

Name: DTQ Input for program AM0020V1 (DQAMTRN1)

Comment: @Definition: DQAMTRN1 is the data queue used to provide input for PGM AM0020V1 which unpackages transactions.

## Process

Hierarchical numeric ID: 1.1.1.1.2.7

Coded name:

Name: DTQ Input for Transmission Validation from Specific trading partner (DQxxxxxx1)

Comment: @Definition: one of several company specific data queues used to provide input to a company specific AM0010V1 PGM used for transmission validation. Data Queues included are DQ\*\*\*011, DQ\*\*\*011, DQVN1011, DQVN2011, DQVN3011, and DQTV4011.

## Process

Hierarchical numeric ID: 1.1.1.1.2.8

Coded name: DQAM70V1

Name: DTQ Input for Time Line Inquiry (DQAM70V1)

Comment: @Definition: DQAM70V1 is the data queue used to provide input to PGM AM0070V1 which provides a support information for oncall work.

## Process

Hierarchical numeric ID: 1.1.1.1.2.12

Coded name: AMLCXLT

Name: PGM Translate Location (AMLCXLT)

Comment: @Purpose: To translate a current rental location to an old location or vice versa to deal with any past "Organization Split" or "Machine Split" conversion projects, such as the "Texas Group Split" of July 1995 or the "Bluegrass Group Split" of June 1997 or the "Group 53 split of October 1999". An example would be calling the program with a TRANSACTION DATE = 08/13/1998, LOCATION CODE = '0' and CURRENT GROUP/BRANCH ID = '5211'. The returned TRANSLATED GROUP/BRANCH ID would be '6511'.

## @Operational Method:

- This program is called with a single input/output AMLCXLT external data structure with the following alphameric data elements and their usage:

ARMS Process Report		
---------------------	--	--

1	(Output)	Return Code
1	(Input)	OLD/NEW CURRENT VALUE
1	(Input)	CURRENT MACHINE ID
2	(Input)	CURRENT GROUP CODE
2	(Input)	CURRENT BRANCH CODE
4	(Input)	CURRENT GROUP/BRANCH
10	(Input)	CURRENT LOCATION
1	(Input)	CURRENT REGION
2	(Input)	CURRENT SUB-REGION
2	(Input)	CURRENT CITY
3	(Input)	CURRENT AREA
20	(Input)	VENDOR TRANSACTION ID
5	(Input)	ARMS PROFILE ID
20	(Input)	CUSTOMER TRANSACTION ID
8	(Input)	TRANSACTION DATE
10	(Output)	TRANSLATED LOCATION
4	(Output)	TRANSLATED GROUP/BRANCH
2	(Output)	TRANSLATED GROUP
2	(Output)	TRANSLATED BRANCH
8	(Output)	TRANSLATED MACHINE NAME
1	(Output)	TRANSLATED REGION
2	(Output)	TRANSLATED SUB-REGION
2	(Output)	TRANSLATED CITY
3	(Output)	TRANSLATED AREA

This pgm will accept a parameter list containing current location information and if a record is found in the ARMS Location Past Conversion Cross-Reference (AMLCKF) file, will return the translation information.

- IF the attempt to retrieve ARMS Rental Transaction Cross-Reference (AMXREF) file record fails, still attempt to perform the translate. Do this by running a test on the passed parameter's CURRENT TRANSACTION DATE.

-- IF all tests for transaction date checking fail, then move 'N' to RETURN CODE.

-- IF a test is successful, perform the routine to load the TRANSLATED MACHINE ID and LOCATION output parameter data elements.

@Notes: This program was written and is currently maintained by the ARMS Application Development Department. It is currently used by the ARMS Unpackage Transaction from Transmission Envelope (AM0020V1 and AM0021V1) programs, along with the ARMS Package Transmission Envelope (AM0120, AM0150, AM0151) programs along with the Financial Systems' AS Account schedules /JOB # 102 (EL0291) program. An interactive stub tester (TSAMLCLT) program exists in production for debugging and testing purposes.

@Files: (CRUD)

AMLCKF (-R--) - Keyed by Old Location ID, along with its other logical access views,

Old Location	AMLCKFL0	- Keyed by
New Location	AMLCKFL1	- Keyed by
Old Group/Branch	AMLCKFL2	- Keyed by
New Group/Branch	AMLCKFL3	- Keyed by
Old Group and Old Branch	AMLCKFL4	- Keyed by

ARMS Process Report		
---------------------	--	--

New Group and New Branch	AMLCKFL5	- Keyed by
Old Group, Region and Sub-Region	AMLCKFL6	- Keyed by
New Group, Region and Sub-Region	AMLCKFL7	- Keyed by
Old Group and City	AMLCKFL8	- Keyed by
New Group and City	AMLCKFL9	- Keyed by
Old Group and Area	AMLCKFL10	- Keyed by
New Group and Area	AMLCKFL11	- Keyed by
Old Group and Region	AMLCKFL12	- Keyed by
New Group and Region	AMLCKFL13	- Keyed by

**Process**

Hierarchical numeric ID: 1.1.1.1.3

Coded name:

Name: AUT Validate Transaction (AM25 - AM46)

Comment: @Definition: The automatic process of transaction validation and database update on the centralized machine prior to dispatch to either the distributed machine or the Rental Management Trading Partner. This process is currently used for both the inbound and outbound processing of transactions.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.1

Coded name: AM0025V1

Name: PGM Validate Transaction Set Record Formats (AM0025V1)

Comment: @Definition: This is the ARMS Transaction Set Record Format Edit never-ending batch ILE RPG program.

**@Purpose:**

The program edits a transmission set group type to ensure that the required formats are present and that no non-allowable record formats exist.

Edit errors from an Inbound (Received) transaction set will be logged and will cause error transaction sets to be generated and sent back to the associated sender.

Edit errors from an Outbound (to be Sent) or Transfer transaction set will be logged and the ARMS On-call support staff will be notified.

**@Operational Method:**

This program endlessly receives the non-keyed input data queue entries from the input data queue (DQAM25V1) that the either of the ARMS Unpackage Remittance/Non-Remittance Envelope programs (AM0021/AM0020V1) generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQAMSET1 data queue and end this program.

<b>ARMS Process Report</b>
----------------------------

For any non-shutdown data queue entry, check the non-shutdown key Group Type Code to be one of the following values:

- Validate Transaction Set Group Type
- Validate Allowable Record Format(s) within Transaction Set for a Group Type

**@Notes:**

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system platform ("RARMS"). Currently, all inbound (received), outbound (to be sent) and transfers are processed in this program as the proprietary EDI Transaction Sets group types.

IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then execute the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's record formats to the output file nor send its data queue entry to the output data queue.

**Called Programs Summary:**

QRCVDTAQ --Read entry from DQAM62V1 data queue  
 QSNDDTAQ --Send entry to DQANDST data queue  
 AMPSSR --ARMS program error handler  
 AM0097V1 --Handle Internal Error  
 AM0090V1 --Generate External Error

@Files: (CRUD)

- AMSET (-R--)
- AMAUTD (-R--)

**@Embedded Data/Constants:**

'\*LIBL' as the DATA QUEUE LIBRARY NAME value used in the execution of the 'QRCVDTAQ' and 'QSNDDTAQ' programs.

'DQAM25V1' and 'DQAMSET1' are the specified data queue names.

'AM0025V1' as the current PROGRAM ID.

'AM0090V1' and 'AM0097V1' are the executed programs, based on the type of error.

'SD' and '\*DOWN' are the shutdown request GROUP TYPE CODE and COMPANY PROFILE ID values.

'OFF' and 'ON'

'I' (inbound), 'O' (outbound), and 'T' (transfer)

ERROR CODES: '16', '45', '69', '70', '71', and '90'.

GROUP TYPE CODES for execution of their associated routine:

<b>ARMS Process Report</b>
----------------------------

'AT', 'EX', 'CN', 'PM', 'ER', 'AK', 'AC', 'IN', 'OF', 'RA', 'RC', 'RE', 'RN', 'CC', 'RM', 'CM', and 'VM'
---

**Compile-Time arrays of:**

- ..Valid transaction set GROUP TYPE CODES
- ..Valid RECORD FORMAT ID's for each transaction set's GROUP TYPE CODE

**@Improvement Opportunities:**

1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') and ARMS Generate External Error ('AM0090V1') programs with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

2.) Delete the constant 'AM0025V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.2

Coded name: AM0030V1

Name: PGM Validate Transaction Set Record Formats Data Types (AM0030V1)

Comment: @Purpose:

To edit a transaction set to ensure the all the required format data elements have values. Whether a data element is required or not is based on the trading partner profile.

**@Operational Method:**

- Wait for entry(s) to exist in DTAQ (DQAMSET1).
- When a shutdown request is received (group type - SD), log the occurrence, pass on the request to DTAQ (DQANDTL1) and wait for the second shutdown.
- When the second shutdown request is received (group type - SD), log the occurrence, pass the request to DTAQ (DQANDTL1) and end.
- When a non-shutdown request is received, read all associated data record(s) from the ARMS Detail Set Transaction File (AMSET) into the appropriate ARMS record format data structures.
- Validate a transaction set's group type record formats' data elements for profilable mandatory values existence.
- Validate data field values with the data format(s) against a domain of allowable values.
- IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then executed the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's record formats to the output file nor send its data queue entry to the output data queue.

**@Notes:**

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system

ARMS Process Report
---------------------

platform ("RARMS"). Currently, all inbound (received), outbound (to be sent) and transfer transaction sets are processed in this program as proprietary EDI Transaction Set group types.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California.

@Files: (CRUD)

-AMSET	(-R--)
-STATE	(-R--)
-AMCLSTBL	(-R--)
-CCSURCHG	(-R--)
-AMERRTBL	(-R--)
-AMXBCOLI	(-R--)
-AMICDTBL	(-R--)
-ARMAP	(-R--)
-ARMSPR1	(-R--)
-ARMSPR4	(-R--)
-AMAUTD	(-R--)

@Embedded Data/Constants:

'0123456789' TO CHECK FOR VALID NUMERIC FIELDS VALUES

'0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ' TO CHECK FOR VALID ALPHANUMERIC FIELDS VALUES

'\*LIBL' as the DATA QUEUE LIBRARY NAME value used in the execution of the 'QRCVDTAQ' and 'QSNDDTAQ' programs.

'DQAMSET1' and 'DQANDTL1' are the specified data queue names.

'AM0030V1' as the current PROGRAM ID.

'AM0090V1' and 'AM0097V1' are the executed programs, based on the type of error.

'SD' and '\*DOWN' are the shutdown request GROUP TYPE CODE and COMPANY PROFILE ID values.

'OFF' and 'ON'

'I' (nbound), 'O' (utbound), and 'T' (ransfer)

All allowable values for each data element

ERROR CODES: '01', '02', '03', '04', '05', '21', '68', '70', '78', and '82'

GROUP TYPE CODES for execution of their associated routine:

'AT', 'EX', 'CN', 'PM', 'ER', 'AK', 'AC', 'IN', 'OF', 'RA', 'RC', 'RE', 'RN', 'CC', 'RM', 'CM', and 'VM'

Compile-Time arrays of:

..Valid transaction set GROUP TYPE CODES

..Valid RECORD FORMAT ID's for each transaction set's GROUP TYPE CODE

ARMS Process Report
---------------------

**@Improvement Opportunities:**

- 1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') and ARMS Generate External Error ('AM0090V1') programs with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.
- 2.) Delete the constant 'AM0025V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.
- 3.) Convert program from OPM RPG to ILE RPG.
- 4.) Replace the use of the AM2010V1 program with the AM1010V1 program called for "I"nquiry.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.3

Coded name: AM0040V1

Name: PGM Validate Transaction Set Record Format Data (AM0040V1)

Comment: @Purpose: To perform the detail edits on any transaction set data element values against the centralized ARMS database and domain of valid values. Also, to assign a rental location to any new Rental Authorization Add transaction that is not authorizing an existing rental reservation or open ticket.

**@Operational Method:**

Wait for keys to exist in DTAQ (DQANDTL1), when key is received, one of three processes occur. Program terminates when group type (SD) shut down data queue entry is encountered and passed on for the second time to output data queue (DQAM46V1).

For a non-shutdown data queue entry, use its value to read all associated records' data values from the ARMS Detail Set Transaction File (AMSET) into the appropriate ARMS record format data structures.

Validate Transaction Set's record formats' data fields' contents for allowable-values.

Validate Transaction Set's functional group type code and its record format's data field's contents against centralized ARMS Rental Transaction Cross-Reference Status and its associated database file fields to ensure that this transaction should be allowed to continue further downstream processing. Send message to ARMS On-Call and reject any Authorization Transfer transaction or outbound transaction to be sent that fails an edit.

IF any received inbound transmission with a specified Branch Claims Office value that does not exist in the Branch Claims Office Cross-Reference file, then assign the first three characters (root customer) of the trading partner's PROFILE ID to ENTERPRISE CUSTOMER ID and send a warning message to ARMS Technical Management (via SNDEMSG) of a transaction with a missing BCO X-Ref record so that they can determine if this BCO is to be added. Do NOT reject the transaction.

ELSE, load the internal routing record format (APPD01) with the retrieved internal ENTERPRISE CUSTOMER ID value.

IF any received inbound transmission with a invalid Date Of Birth value, then set the value to zeros and send a warning message "Invalid DOB" to ARMS Technical Management which is currently not acted upon by them.

**ARMS Process Report**

The messages sent to ARMS Technical along with the employee numbers to send the message to exist in file SNDMSG.

IF an authorization-add transaction set or a group-to-group authorized reservation transfer transaction set has a non-blank rental location (Group ID and Branch ID), then validate the rental location against the Daily Rental Office Locations (DROFF) file by the Rental Location ID. IF not found for the authorization-add/transfer, or, for an authorization-add, the location found indicates that it does prevent ARMS reservations, then use its non-blank Forward ARMS Reservations To Location. If the Forward ARMS Reservations To Location is blank or not found, then derive the closest available daily rental location and its platform.

IF a received unsolicited Authorization Add rental transaction set is received that is to authorize and create a currently non-existent rental reservation, derive and assign closest available rental location and its distributed application host platform for later routing to that platform.

To derive the closest available rental location we use RAS013A.

IF the authorization add/transfer transaction set is for Enterprise Fleet Management VIP rental customer, they will be able to send and generate ARMS authorized reservations to airport branches if that is the closest rental location. Any other trading partner authorization add/transfer cannot be routed to an airport location.

Derive that rental location's associated distributed application host platform by executing Derive Machine ID for Rental Location (AM2050V1) program for later routing.

IF a transaction's record formats are validated and processed successfully, then send its associated key value as output to the data queue (DQAM46V1) that is input to the next downstream process for ARMS Database Update and Routing (AM0046V1) program.

@Files: (CRUD)

AMSET	(CRU-)
ARMSPR1	(-R--)
AMAUTD	(-R--)
AMIEBT	(-R--)
AMXBCO	(-R--)
DROFLF1	(-R--)
DROFLF7	(-R--)
FILE80	(C---)

@Notes:

This program is submitted for execution as a never-ending batch program/job in the ARMS subsystem by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the centralized ARMS host system platform ("RARMS"). It will usually be active from 3:00am to 1:00am-Central. Currently, all inbound (received), outbound (to be sent) and transfer transaction sets are processed in this program as proprietary EDI Transaction Set group types.

IF any error is found, then do not process this transaction any further and if it is an inbound transaction set, then execute the ARMS Generate External Error (AM0090V1) program, ELSE, execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded ERROR CODE value. Do NOT output the transaction set's



### ARMS Process Report

record formats to the output file nor send its data queue entry to the output data queue.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California.

#### @Embedded Data/Constants:

'7680       ' used as the default Claims Connection Rental Location.  
 '6A01       ' used for routing to this simulation default rental location within the production software environment.  
 '0123456789' used as valid numeric values in a CHECK operation.  
 'RTVMSGSTS' is the program executed to check if the e-Message System is available for sending messages.

'OVRPRTF FILE(QPPGMDMP) OUTQ(ARMSDUMP) OUTPTY(9) USRDTA(AM40PGMERR) FORMTYPE(ARMS) HOLD(\*YES)' to cause generated RPG-formatted dump output to be retained for greater than a one day.

The following are 25-character ERROR MESSAGES subject header and detail line prefixed output to FILE80 for sending when executing MS0010 program:

'An invalid BCO was sent'  
 'Invalid BCO'  
 'Profile'  
 'transaction control ID'  
 'Invalid DOB was received'  
 'Invalid DOB'

#### @Improvement Opportunities:

- 1.) Remove as constant and store the default simulation rental location ID ('6A01') and the default production Claims Connection rental location ID in an external data area to externalize this location information for inquiry into the program.
- 2.) Should be able to replace the valid numeric values constant and its usage with an ILE built-in function or existing API.
- 3.) Should be able to duplicate the existing OS/400 Dump printer file QPPGMDMP to a different ARMSDUMP printer file that is formatted the same but has all of the specified keyword attributes that would allow the removal of the OVRPRTF OS/400 command execution.

#### Process

Hierarchical numeric ID:       1.1.1.1.3.4  
 Coded name: AM0046V1  
 Name: PGM Route Transaction / Update ARMS Centralized Database (AM0046V1)  
 Comment:   @Purpose: To update the centralized ARMS database with transaction set data elements and forward the transaction set to the appropriate destination (trading partner, rental application, claims connection, or centralized financial system's cash receipts interface).

#### @Operational Method:

### ARMS Process Report

- Maintain centralized ARMS application system database with input transaction set data elements.
- Route outbound transaction data set for external envelope packaging.
- Route inbound or transfer transaction data set for sending to distributed host platform.
- IF processing an RA Change, AT Change or an an Opening Rental Notification RN functional group type, then calculate the renter's age by subtracting their Date of Birth from the system date. IF the renter is underage for the state of the rental and the ticket is open and authorized, then execute the Generate Surcharge subroutine (GENSUR) to load the Approved Surcharge Detail (AMSURD) file. This subroutine calls AM1085V1 for underage surcharges and AM0096V1 for othe surcharges. An RA-Change for an ARMS Trading Partner Company will only be generated if the transaction has been confirmed. (See @Notes for more detailed information)
- On an open authorized ticket, if a different Rental Location ID is processed from an outbound or transfer transaction set, then execute the Add Government Surcharge (AM0096V1) program to load the Approved Surcharge Detail (AMSURD) file.
- IF any new approved surcharges were generated, then read this file for this rental transaction to generate a new automatic pseudo-rental Authorization Change (AT) transaction set for approved surcharges to be routed to a distributed rental application system host platform.
- IF a Customer Extension/Termination, functional group type EX is received from a Trading Partner Company for a defined Profile ID AND the BCO or Adjuster Code is different than on the ARMS centralized database, then generate a new automatic pseudo-rental Authorization Change (AT) transaction set containing only an adjuster detail (ADJD01) record format for an adjuster/BCO change to be routed to a distributed rental application system host platform. This is done to update Bill-To Customer information in the rental system database files for this rental transaction in the EC00ATV1 program processing, since the EC00EXV1 program processing does not update Bill-To Customer information in the rental system database files. (See @Improvement Opportunities number 3 below.)
- IF processing an Extension, move an "E" to the Action Code in the AM055V1 external data structure. When a termination date has been sent, update the Day Auth (AWAUDY) field in file AMAUTD.  
Calculate the total number of days authorized by using the Termination Date and Rental Start Date adding 1 to the calculated days to receive the correct number of days authorized. If the calculated amount is greater than the Day Auth field, move an "E" to the Action Code in the AM055V1 external data structure...  
If the calculated amount is less than or equal to the Day Auth field, move a ~~TT~~ to the Action Code in the AM055V1 external data structure.
- Based on the trading partner profile, generate new automatic rental Extension days transaction set in response to a Request for Extension (RE) to be routed to a distributed rental application system host platform.
- IF processing an Opening Rental Notification ('RN') functional group type, IF the existing customer number on the transaction set internal header is found in the ARMS BCO for RMS (Rental Management System) file write a record to the ARMS RMS (Rental Management System) File (AM002P) so that RMS will be notified of the open rental contract/ticket that they must manage from their centralized location.
- IF processing a Rental End ('RN') transaction set AND this is the first closing rental notification (AMAUTD's Rental End Date = 0) AND IF the associated

### ARMS Process Report

ARMSPR1 profile file record's Remittance Indication Flag (PIFG14) field is equal to 'A', then execute the ARMS Generate Transaction Credits (AM1080V1) program passing the Transmission Control Sequence ID Number, Group Control Sequence ID Number, Vendor Transaction ID and the Execution Code of 'R' (for Rental Notification).

- IF processing a Payment Advice ('PM') transaction set, then execute the ARMS Generate Transaction Credits (AM1080V1) program passing the Transmission Control Sequence ID Number, Group Control Sequence ID Number, Vendor Transaction ID and the Execution Code of 'P' (for Payment Advice).

-After the file updates and before the next data queue entry is read, and if the Group Type is not ~~OF~~.

-Complete the build of the data queue entry record.

-From the APPD01 format, move the Profile Id, Customer Transaction Id, Vendor Transaction Id, -Group Type, Date, and Time to their appropriate fields in DQAM55V1 external data structure, AM055X.

-If Transaction Type (AM055X) is blank, move an ~~id~~ internal to the field.

-Send the data queue entry to DQAM55V1.

-Clear the AM055X format for the next record.

#### @Notes:

Currently, ALL transaction sets (received from trading partners, received from rental systems - i.e., inbound, outbound and transfers) will be processed by this program, after editing for updating to the database and routing (to a trading partner or a rental application system).

When the Rental Management Trading Partner sends an extension transaction set for an existing Vendor Transaction ID with a different Customer Transaction ID than the value in the ARMS Cross-Reference (AMXREF) file record's CUSTOMER TRANSACTION ID (XACTID) field, then this field change should occur and allow the AMXREF file record's XACTID field to be updated with the new/different Customer Transaction ID value. All incoming transaction sets should allow a change of the Customer Transaction ID. Remittance Advice ('RM') transaction sets will not be included because this would require a change in the way RM's are processed.

When a request for cancellation ('RC') transaction set is processed for an incompletely open rental contract/ticket (prewrite/unit-pend), this 'RC' transaction set is not sent to the trading partner. In most cases this is correct, however when the ARMS Cross-reference file record is made irretrievable by padding the rental location value with the repeating 'X' character, the request for cancellation transaction set will be sent. For some profiled ARMS Trading Partner Companies, the RC is converted to a CN before being sent.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer); and AAA of California. Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

When a Request for an Authorization is sent to ARMS Trading Partner Company after the initial request, and the transaction is still in an unconfirmed status (U), an RA-Add will be generated. IF the transaction is at a confirmed status (W) but still unauthorized an RA-Change will be generated even if no changes have

ARMS Process Report
---------------------

been made to the data. If the transaction is at a confirmed and Authorized status (R), an RA-Change will be generated only if there was a change in the data.

@Files: (CRUD)

```

- AMSET          (-R--)
- ARMSPR1        (-R--)
- ARMSPR3        (-R--)
- AMXBCO         (-R--)
- AMPGMCTL       (-R--)
- AM002P         (CR--) ARMS: RMS FILE
- AM003P01       (-R--) ARMS RMS BCO FILE
- AMAPPS         (C---)
- AMPACK         (C---)
- AMADJD         (CRU-)
- AMAUTD         (CRU-)
- AMCOMD         (CR--)
- AMIEBD         (CRU-)
- AMIEBH         (CRU-)
- AMIEBT         (CRU-)
- AMINSO         (CRU-)
- AMRMTD         (CRU-)
- AMRNTD         (CRU-)
- AMRPRD         (CRU-)
- AMRATD         (CRUD)
- AMSURD         (CR-D)

```

@Embedded Data/Constants:

```

'YES'
'NO '
'AM0046V1 '
'RECORD DELETED'
'XXXXXXXXXX' (to pad ARMS Rental Transaction Cross-Reference File
record's RENTAL LOCATION with X's to make it irretrievable)
'AUTHORIZATION EXPIRED' (as COMD01 record format's COMMENT comment text
currently not referenced, unsure of intent.)

```

Enterprise Customer Numbers for ARMS Trading Partner Insurance Company-  
Los Angeles Area Locations and ARMS Trading Partner Insurance Company's Branch Claims  
Office Code for RMS output:

```

'ALL3233' and '2040' Santa Ana/Tustin, California
'ALL3221' and '1240' Torrance, California

```

.....and Format Names/IDs::

```

'APPD01'
'ADJD01'
'ADJD02'
'AUTD01'
'CAND01'
'CBKC01'
'CBKD01'
'CRED01'
'COMD01'
'ELCD01'
'EERD01'
'IEBD01'
'IEBH01'

```

**ARMS Process Report**

'IEBT01'  
'INSD01'  
'OFFD01'  
'PMTD01'  
'RATD01'  
'REND01'  
'RMTD01'  
'RNTD01'  
'RNTD02'  
'RPRD01'  
'SURD01'  
'VEDD01'

**@Improvement Opportunities:**

1.) Convert from standard ILE RPG program to an ILE program with multiple ILE RPG modules or service programs.

2.) Delete the 'AM0046V1 ' constant and use the Program Status Data Structure's left-justified PROGRAM ID value for the same purposes.

3.) There is hard-coded logic to check IF ARMS Trading Partner Insurance Company Insurance Company customer authorization extension ('EX') transaction set AND the adjuster or BCO change that is different than the centralized ARMS rental transaction database Adjuster Detail (AMADJD) file data, then to automatically generate a "pseudo-" Authorization ('AT') Change transaction set with only an adjuster detail (ADJD01) record. However, since this is information that is only for a specific X12 EDI standard compliant trading partner, that this special logic should be removed from this program and externally profiled in the EDI Translation and Interface/Mapper software to cause generation of the additional 'AT' Change when an Extension/Termination transaction set is received. Additionally, this EDI Translation and Interface Mapper software could execute an ARMS centralized database retrieval inquiry service utility program to retrieve specified information from the AMADJD file. Then this interface/mapper software could determine if this transaction set's adjuster information is different than the existing database file's information and if true, generate the additional 'AT' Change. This would add an 'AT' Change functional group transaction set to be output, along with the 'EX' from the same EDI transmission envelope, and it would add to the transaction load between AM0010V1 and AM0046V1 programs, inclusive. However, the benefit of standardizing the transaction set processing within AM0046V1 and moving profile-specific logic to the EDI translation interface/mapping software would be greater than having hard-coded within AM0046V1 program.

4.) There is currently an undesirable situation that occurs within this program. When any trading partner sends an authorization extension/termination transaction set, the adjuster detail is required to add callback detail note comments to the rental application database with the name of the adjuster, changing the open rental contract/ticket's CURRENT AUTHORIZATION DATE field value. However, the adjuster making this change in this value may not be the claims adjuster assigned to manage the direct billing payment authorization for this rental contract. The adjuster making the extension/termination may be another adjuster taking on the extensions temporarily when the primary assigned adjuster is unavailable (i.e., day off due to vacation, illness, etc.), but that the primary assigned claims adjuster "of record" was not intended to be changed. Currently, the adjuster information in the Bill-To information within the rental contract is not updated by an extension/termination transaction set's adjuster information, but the centralized ARMS database adjuster detail (AMADJD) file is. This is erroneous to update AMADJD by an extension and this logic should be removed from AM0046V1 program. (A new issue will be submitted by Gina Miller, ARMS Technical Mgmt. staff to request this change.

**ARMS Process Report**

However, if AM0046V1 program's extension/termination logic is revised, then this issue should be considered.)

5.) Drop the constant detailed below and replace with attributes/field values within the ARMS Trading Partner Profile's Branch Claims Office / Enterprise Customer ID Cross-Reference (AMXBCO) file. Likewise, add the maintenance of these fields to the AMXBCO maintenance program and screen display file record format (AAAM10).

**Process**

Hierarchical numeric ID: 1.1.1.1.3.5

Coded name:

Name: DTQ Input to AM0025 (DQAM25V1)

Comment: - @Definition: DQAM25V1 is a data queue used to provide input for PGM AM0025V1 which validates transaction set record formats.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.6

Coded name:

Name: DTQ Input Data Queue to AM0030 (DQAMSET1)

Comment: @Definition: DQAMSET1 is a data queue used to provide input for PGM AM0030V1 which validates transaction set record formats data types.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.7

Coded name:

Name: DTQ Input Data Queue to AM0040 (DQANDTL1)

Comment: @Definition: DQANDTL1 is a data queue used to provide input to PGM AM0040V1 which validates transaction set record format data.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.8

Coded name:

Name: DTQ Input Data Queue to AM0046 (DQAM46V1)

Comment: @Definition: DQAM46V1 is a data queue used to provide input to PGM AM0046V1 which update the ARMS database and routes the transaction.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.9

Coded name:

Name: DTQ Input to Program AM0100 (DQAMAP1)

Comment: @Definition: DQAMAP1 is a data queue used to input data to PGM AM0100 which distributes routed transactions from the centralized host processing machine.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.10

Coded name: AMPSSR

Name: PGM Perform Internal Error Paging and Messaging (AMPSSR)

### ARMS Process Report

**Comment:** @Definition: This is the ARMS \*PSSR program exception error paging & messaging program (written in RPG/400 - Original Program Model).

**@Purpose:** This program's purpose is to output the error occurrence to the AMPGMERR Program Error Log file and if the error is not from a program executing on the "DEV" (development) machine, then to notify the ARMS On-Call and Backup persons via sending a pager message and sending an 'e-message' via call to MS0010 program of the platform name, program name and the error occurrence. This information will aid the ARMS On-Call person responding to resubmit any needed never-ending program job into the ARMS subsystem on the appropriate host computer system. This information will also cause them to research the dump taken in the previous program for the incompletely processed transaction specific information that it had been processing when the program halted or abnormally ended.

**@Operational Method:** This program outputs the passed entry parameters' values error occurrence to the AMPGMERR Program Error Log file. If the error is not on the "DEV" (development) host platform, then sending a pager message to the ARMS Primary and Backup On-Call pagers and sending an 'e-message' via call to MS0010 program of the platform name, program name and the error occurrence.

**@Files:** (CRUD)

- AMPERT (-R--) , accessed by AMPERTL1 logical access view.
- ANDQER (-R--) , accessed by ANDQERL2 logical access view.
- AMPGMERR (C---)
- FILE80 (C---)

**@Notes:** This program is used within every program's \*PSSR program exception subroutine.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.11

Coded name: CCDATE

Name: PGM Validate Date and Convert Format (CCDATE)

**Comment:** @Definition: CCDATE is a date validation, conversion and manipulation callable OPM RPG program.

**@Purpose:** Executed from many ARMS programs for many purposes. The most common purposes are: to validate dates; convert formats; derive a future date by computing the adding provided days to a given date value; or, to derive the age of a renter by computing the number of years between the current date and a renter's date of birth. In a majority of programs, this program is executed to retrieve the current system date or timestamp in the MMDDYYYY or MMDDYY format and reformat-it to YYYYMMDD format, or, to retrieve the day of week, such as required in the ARMS-to-ARMS Trading Partner Company Communication Sender / Receiver (AM\*\*\*COM) program.

Validation is done throughout many ARMS programs, such as in the Validate Transmission Envelope (AM0010V1) program to validate the Transmission Date or in the Validate Transaction Set Record Format Data Types (AM0030V1) program's Edit Dates subroutine specifically to validate the date values for the following record formats' date value fields:

Authorization Detail's (AUTD01) Authorization Expiration Date, Date of Birth and Pickup Date

Renter Detail's (RNTD02) Date of Loss

Insured Detail's (INSD01) Policy Expiration Date

Cancellation Detail's (CAND01) Cancellation Date

Callback Detail's (CBKD01V1) Callback Origination Date

ARMS Process Report
---------------------

Extension/Termination Detail's (CRED01) Termination Date and Renter  
 Notified Date  
 Rental Detail's (REND01) Rental Start and End Dates  
 Rental Segment Detail's (RATD01) Effective Dates of Rental Segments 1  
 through 4  
 Invoice Header's (IEBH01) Billing Start and End Dates  
 Payment Detail's (PMTD01) Payment Date

**@Operational Method:**

Before the CCDATE is called, the parameter fields must be set by the calling program. It is recommended that you do a clear the DTPARM external data structure subfields before any field values are loaded to insure that all fields have been initialized.

If a date conversion from MMDDYY to YYMMDD format, simply move the input date value to the From-Date (DTFRDT) input field after all the fields have been cleared and execute the CCDATE program. The outputted To-Date returned in DTTODT will be in YYMMDD format.

The following fields will be output by CCDATE, so it is not necessary that you fill them in. They are: DTFMTD, DTWKDY, DTMNTH, DTSTNG, DTMSID, and DTMSTX.

DTDAYS is not necessary for action code 'C' and will be zeroed out by CCDATE. DTDAYS will be filled in by the CCDATE program's execution when the action code is 'D'.

However, since AM0030V1 is requiring only date value validation, the program is executed after loading the following parameter subfield values in the following manner after all subfields were cleared/initialized:

Load 'C' (Convert) to ACTION CODE (DTACCD)  
 Load the DATE TO VALIDATE (in YYYYMMDD format) field value to FROM-DATE (DTFRDT)  
 Load the number 7 (YYYYMMDD) to the FROM-DATE-FORMAT (DTFRFT)  
 Load the number 6 (DDMMYYYY) to the TO-DATE-FORMAT (DTTOFT)  
 Execute the program  
 Check the contents of the Error Message ID (DTMSID). IF not blanks, then the FROM-DATE field value was invalid.

**@Notes:**

Documentation for usage can be found on the AS/400 in the ELLIB/QEISSRC/TLS.DAT text source member.

NEVER Use the DATE MULTIplied. by 1000.01 or 100.0001 to CONVERT! (Lack of precision and high overhead mathematical computation overhead.)

It should do hopefully everything anyone would need from such a program for any date value. Below is the explanation of its single parameter and how to use the program.

The parameters have been set up as in a empty database file called DTPARM that is used as an externally defined data structure on an input specification on the program's source code that causes the compiler to copy the subfield names, attributes, and starting positions. This is done so that a programmer can copy the parameters in as an external data structure instead of having to remember what order the parameters should be in and what length the fields are. The explanation of the external data structure DTPARM parameter data elements (subfields) passed as a single parameter field are:



<b>ARMS Process Report</b>
----------------------------

Convert (DEFAULT) DTACCD - (1) - Processing Action Code possible values: 'C' -

'D' - Compute the Days Difference between inputted dates

'A' - Add inputted Days to inputted From-Date

DTFRDT - (8,0) - From-Date (DEFAULT is UDATE)

DTFRFT - (1,0) - From-Date Format

possible values: 1

- MMDDYY (Month, Day, Year) (DEFAULT)

2 - DDMYY (Day, Month, Year)

3 - YYMMDD (Year, Month, Day)

4 - YYDDD (Julian format)

5 - MMDDYYYY

6 - DDMYYYY

7 - YYYYMMDD

8 - YYYYDDD (Julian format)

9 - MM00YY (Use this format to have the program determine the last day of the specified

month and format it based on the To format.)

\*\*NOTE: This format is only valid for this From-format:

DTODT - (8,0) - To-Date

DTOFT - (1,0) - To-Date Format

possible values: 1

- MMDDYY (Month, Day, Year)

2 - DDMYY (Day, Month, Year)

3 - YYMMDD (Year, Month, Day) (DEFAULT)

4 - YYDDD (Julian format)

5 - MMDDYYYY

6 - DDMYYYY

7 - YYYYMMDD

8 - YYYYDDD (Julian format)

DTDDAYS - (7,0) - Number of Days to Add/Subtract or the computed Days Difference (+/-)

DTFMTD - (10) - Displayed Formatted Date

DTSEP - (1) - Display Separator (used in DTFMTD, i.e., backslash, hyphen, or period)

DTWKDY - (9) - Day of Week (Sunday, Monday, etc.)

DTMNT - (9) - Month Name (January, February, etc.)

DTSTNG - (30) - Formatted Date String

possible values:

'C'  
1992)

- Month Day, Year

(i.e., January 01,

'F' (DEFAULT) - Day-of-week Month Day, Year (i.e., Wednesday January 01, 1992)

**ARMS Process Report**

DTMSID - (7) - Error Message ID  
DTMSTX - (80) - Error Message Text

**@Improvement Opportunity:**

The execution of this (CCDATE) program would be unnecessary if all of the ARMS programs that currently use this program were converted from the OPM RPG/400 to ILE RPG/IV program and these input date fields were attempted to be loaded into date type work fields. Also, any unsuccessful operation would need to be monitored for specific error exception types.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.12  
Coded name: AM2010V1  
Name: PGM Retrieve Cross-reference (AM2010V1)  
Comment: @Definition: This program is the callable ARMS generic Cross-Reference File Record Inquiry program.

@Purpose: To retrieve the associated ARMS Rental Cross-Reference File record when passed one of the following four input combinations: the ARMS VENDOR TRANSACTION ID; the ARMS TRADING PARTNER COMPANY PROFILE ID and the CUSTOMER TRANSACTION ID; the RENTAL LOCATION and the RENTAL CONTRACT/TICKET ID; or, the RENTAL LOCATION and the RENTAL BRANCH RESERVATION ID.

**@Operational Method:**

1.) The RENTAL LOCATION is always required and either a non-blank TICKET ID or RESERVATION ID must be passed.

IF RENTAL LOCATION, TICKET ID and RESERVATION ID are blanks, then load the RETURN CODE with 'E' (Error).

2.) IF Rental Location and Reservation ID are passed with the program call, use AMXREFL2 to retrieve the requested ARMS Cross-Reference File record.

a.) IF Ticket number is also passed with the program call, retrieve the record using AMXREFL2. Then validate the Ticket number passed with the program call against the Ticket data field XATKNO.

b.) IF passed TICKET ID and retrieved TICKET ID are different, then load the RETURN CODE with 'T' (Ticket ID Mismatch) and return to the calling program.

c.) IF the passed TICKET ID is same as the TICKET ID found for the reservation in the ARMS Cross-Reference File, then update all fields in the parameter list from AMXREFL2, set the RETURN CODE to blanks (blanks indicate successful) and return to the calling program.

3.) IF non-blank RENTAL LOCATION and TICKET ID are passed with the program, use AMXREFL1 to retrieve the data.

IF no record found for RENTAL LOCATION and TICKET ID, then load the RETURN CODE with 'E' (Error) .

4.) If an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with XACUID in AMXREFLn.

IF they are different, change the RETURN CODE to 'P' for 'Profile Error' and return to calling program.

<b>ARMS Process Report</b>
----------------------------

5.) IF non-blank ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, retrieve record from AMXREFL3.

a.) IF TICKET ID / RESERVATION ID is passed, compare these with values retrieved from AMXREFL3.

IF they do not agree, change return code to 'T' / 'R'.

b.) IF RENTAL LOCATION is passed, compare this value with value retrieved from AMXREFL3.

IF they do not match, change return code to 'L' .

6.) IF an ARMS VENDOR TRANSACTION ID is passed, retrieve record from AMXREF.

a.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with the associated field in the retrieved ARMS Cross-Reference File.

IF they are different, change the return code to 'P' for 'Profile Error' and return to calling program.

b.) IF RENTAL LOCATION is passed, compare this value with the AMXREF value. IF they are different, change return code to 'L'.

c.) IF TICKET ID or RESERVATION ID is passed, compare these values with values retrieved from AMXREF.

IF they do not agree, change RETURN CODE to 'T' / 'R'.

7.) IF a record is retrieved from AMXREF or any of these logical files based upon the file use hierarchy, and additional data fields are passed, and more than one data mismatch occurs, change the return code to 'M' for multiple errors.

#### Hierarchy for record retrieval:

1.) IF ARMS VENDOR TRANSACTION ID is passed, and any other data fields, retrieve from AMXREF.

2.) IF ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, and any other data fields (except VENDOR TRANSACTION ID), retrieve from AMXREFL3.

3.) IF RENTAL LOCATION and RESERVATION ID are passed, retrieve from AMXREFL2.

4.) IF RENTAL LOCATION and TICKET ID are passed, retrieve from AMXREFL1.

#### RETURN CODE possible values:

```

-----
blank : Successful retrieval / edit
L : Location passed is blank / invalid
# : Ticket/Reservation passed is blank
T : Ticket passed, unmatched on AMXREF record
P : Profile passed, unmatched on AMXREF record
E : No AMXREF record found
M : Multiple errors
  
```

# ARMS Process Report

@Notes: No exception error handling is specified.

This executable program is executed by many other ARMS programs that are processing a ARMS transaction request to retrieve information from the ARMS Rental Cross-Reference File. This program is currently called with parameters by the following programs: AMLCXLIT, AMRTVSQ, AM0020V1, AM0021V1, AM0030V1, AM0040V1, AM0070V1, AM0120, AM0129, AM0150, AM0900, AM1080V1, AM1085V1, AM5010, AM5015, and CCAM16. The parameters used in calling this program are the following:

	Output	1 character	RETURN CODE
	Input/Output	20 character	VENDOR TRANSACTION ID
	Input/Output	5 character	ARMS TRADING PARTNER COMPANY PROFILE
ID			
	Input/Output	20 character	CUSTOMER TRANSACTION ID
	Input/Output	10 character	RENTAL LOCATION
	Input/Output	6 character	RESERVATION ID
	Input/Output	6 character	TICKET ID
	Output	1 character	STATUS CODE
	Output	8/0 numeric	STATUS CODE LAST CHANGED DATE
	Output	6/0 numeric	STATUS CODE LAST CHANGED TIME
	Output	10 character	STATUS CODE LAST CHANGED PROGRAM
ID			
	Output	1 character	MACHINE ID
	Output	10 character	BRANCH CLAIMS OFFICE CODE
	Output	1 character	RENTAL SYSTEM SOURCE ID
	Output	8/0 numeric	RECORD ADD DATE
	Output	6/0 numeric	RECORD ADD TIME
	Output	10 character	RECORD ADD PROGRAM ID
	Output	8/0 numeric	RECORD CHANGE DATE
	Output	6/0 numeric	RECORD CHANGE TIME
	Output	10 character	RECORD CHANGE PROGRAM ID

@Files: (CRUD)

- AMXREF (-R--) ARMS Cross-Reference File, keyed by Vendor Transaction ID

- AMXREFL3 (-R--) ARMS Cross-Reference File, keyed by Company Profile ID and Customer Transaction ID

- AMXREFL1 (-R--) ARMS Cross-Reference File, keyed by Rental Location ID and Rental Ticket ID

- AMXREFL2 (-R--) ARMS Cross-Reference File, keyed by Rental Location ID and Rental Reservation ID

@Improvement Opportunities:

Replace the internally defined parameter fields in this program and all programs executing this program with a database file with no record to be used as an externally defined data structure so that only one definition of the parameter's data elements exist, along with comments on their purpose and possible values.

Replacing the execution of this (AM2010V1) program with the ARMS Inquire/Update Cross-Reference File record (AM1010V1) program in many other programs if it is called in "Inquiry" mode.

Process

Hierarchical numeric ID: 1.1.1.1.3.13

<b>ARMS Process Report</b>
----------------------------

Coded name: AM0090V1

Name: PGM Generate External Error (AM0090V1)

Comment: @Definition: This program is the ARMS called External Error Handler Program.

@Purpose: To generate a single Error (ER) group type transaction data set to be sent to the Rental Management Trading Partner to indicate that the ARMS host rejected a transaction data set received from their host application system. The error is also logged to an error log file.

## @Operational Method:

- Verifies the passed input parameter error reason codes (maximum 20) that are not equal to '99' against the ARMS Error Code and Description Table (AMERRTBL) file for valid error codes and returns an "A" for accepted or an "R" for rejected. If the passed error reason code does not exist in this file, the ARMS Program Exception Handler (AMPSSR) program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the text "ERROR CODE IS INVALID" and this resulting value is written to the ARMS Error Log (AMERRLOG) file record's FILLER field and its ERROR CODE field is loaded with the value "99".

- Write Error (ER) group type transaction data set records to one of two possible transaction data set output files. If the program is being executed on the centralized ARMS host computer system platform, it will write the output record formats to the ARMS Transactions To Be Packaged (AMPACK) transaction file as centralized transactions to be packaged for sending. If this program is executing on the distributed rental computer system platform, then it will write the output record formats to the ARMS Distributed Transactions To Centralized Input (ANDIST) transaction file for the distributed system transaction to be sent to the centralized system.

- Sends the written Error group type transaction set record formats' key value to the output file's associated data queue. If the records were output to the AMPACK file, then the key is sent its associated Transaction Packaging Input (DQAMPKG) data queue. If the records were output to the ARMS (ANDIST) file, then the key is sent its associated Transaction Distributed-to-Centralized Input (DQANDST) data queue.

- Before exiting the program, a record is also added to rejected transactions error log file (AMERRLOG) that will appear on the ARMS Error Log Report (CLL817/AM5000) and the ARMS Error Log Inquiry Screen (CCAM14).

@Notes: This program is only executed by other ARMS never-ending programs that have validated a received transaction data set as being in error and rejected it from further processing to eventually update the Rental Systems application database. This program is currently called with parameters by the following programs: AM0020V1, AM0021V1, AM0025V1, AM0030V1, and EC00EXV1. (NOTE: This program has been replaced in many other programs that it had been called from by calling the newer ARMS Internal/External Error Handler 'AM0098' program.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR

<b>ARMS Process Report</b>
----------------------------

Input 40 character code/20 occurrences maximum)	ERROR REASON CODE(S) (2 characters per
Output 1 character R=Rejected - default, or A=Accepted)	PROGRAM RETURN CODE (Possible values:
Input 10 character transaction processing)	PROGRAM NAME IN ERROR (Rejecting the

@Files: (CRUD)

- AMPACK (C---)
- ANDIST (C---)
- AMERRTBL (-R--)
- AMERRLOG (C---)

@Embedded Data: (Constants)

"ERROR CODE IS INVALID" as suffix for invalid error code.

@Improvement Opportunity:

Since the text is the same in all of the files, move the constant's error message text into an added record in the AMERRTBL file with the ERROR CODE value of blanks that this program, AM0090V1 Generate ARMS External Error and AM0098 ARMS Error Handler program could all retrieve.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.14

Coded name: AM0097V1

Name: PGM Handle Internal Error (AM0097V1)

Comment: @Definition: This program is the ARMS called Internal Error Handler Program.

@Purpose: To notify the ARMS Application Development Department On-Call personnel of a transaction error condition that requires research and possible manual intervention. Error (ER) group type transaction data set in response to a rejected transaction data set received from a Rental Management Trading Partner host application system and to log the error to an error log file.

@Operational Method:

- Verifies each non-blank passed input parameter error reason code that is not equal to '99' or '00' exists in the ARMS Valid Error Codes table file (AMERRTBL) and returns an 'A' for accepted or an 'R' for rejected. If the passed error reason code does not exist in this file, the ARMS Program Exception Handler (AMPSSR) program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the constant text "ERROR CODE IS INVALID" and this resulting value is written to the ARMS Reservation Transfer (and Outbound) Errors Log (AMKFRLOG) file record's FILLER field and its ERROR CODE field is loaded with the value '99' (Unknown).

- If the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to the ARMS Program Exception Handler 'AMPSSR' program, passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

@Note: This program is only executed by other ARMS never-ending programs or called programs that cannot process a ARMS transaction request as being in error

ARMS Process Report
---------------------

or not doable and rejected it from further processing. This program is currently called with parameters by the following programs: AM0025V1, AM0030V1, AM0047V1, AM0062V1, AM1080V1, AM2090V1, CC2090V1 and EC2090V1. (NOTE: This program has been replaced in many other programs that it had been called from by calling the newer ARMS Internal/External Error Handler 'AM0098' program.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR
Input	40 character	ERROR REASON CODE(S) (2 characters per
code/20 occurrences maximum)		
Output	1 character	PROGRAM RETURN CODE (Possible values:
R=Rejected - default, or A=Accepted)		
Input	10 character	PROGRAM NAME IN ERROR (Rejecting the
transaction processing)		

@Files: (CRUD)

- AMERRTBL (-R--)  
- AMXFRLOG (C---)

@Embedded Data: (Constants)

"ERROR CODE IS INVALID" as suffix for invalid error code.

@Improvement Opportunity:

Since the text is the same in all of the files, move the constant's error message text into an added record in the AMERRTBL file with the ERROR CODE value of blanks that this program, AM0090V1 Generate ARMS External Error and AM0098 ARMS Error Handler program could all retrieve.

## Process

Hierarchical numeric ID: 1.1.1.1.3.15

Coded name: CRTF80

Name: PGM Create MS01 (CRTF80)

Comment: Definition: This is the executable OPM CL program to Create and Clear the temporary Send "e-Messages" Work (FILE80) Message Detail Lines file.

@Purpose: To duplicate, if necessary, and clear a temporary output-work file prior to outputting "e-message" detail message lines from an executing program to be used as input to a subsequent execution of the Send "e-Message" (MS0010) program.

@Operational Method:

Checks the existence of a ISS: SM0003 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library.

IF the file does not exist, then retrieve the library from this file from any library within this job session's library list. Then create a duplicate copy of this file in the QTEMP Temporary Job Session library without copying any data within the file.

**ARMS Process Report**

Clear the 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library.

Execute an override of any specified 80-position work file (FILE80) in this active job session for the scope of the entire job to 80-position work file (FILE80) in this active job session's QTEMP Temporary Job Session library and sequentially block its access with 409 records per block.

**@Notes:**

This program was created and is stewarded by the Information Systems Support Department's ISA Group.

No parameters are used in the execution of this program.

**@Files:** (CRUD)

FILE80 (----) File is created in the QTEMP library if necessary and then cleared.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.16

Coded name: RTVMSGTS

Name: PGM Retrieve Message Status (RTVMSGSTS)

Comment: @Definition: This is an executed OPM CL program to retrieve the status of the Enterprise Rent-A-Car Message Application System.

@Purpose: To retrieve the status of the Enterprise Rent-A-Car Message Application System before execution of the Send a Message (MS0010) program.

**@Operational Method:**

Load the default value 'ER' (Error) to STATUS CODE.

IF currently executed on the "DEV" machine, then retrieve production objects from ELPRD library instead of the ELLIB library.

Retrieve the logical value of the MESSAGE SYSTEM STATUS from its Status data area (DAMSGSTS).

IF the MESSAGE SYSTEM STATUS is "ON" ('1' or True), then the MESSAGE SYSTEM is UP. Then check the DDM and OPTICONNECT/400 STATUS values on the 'OPTICON' machine/system platform. IF both are OK, then pass 'OK' to STATUS CODE.

IF the MESSAGE SYSTEM STATUS is "OFF" ('0' or False), then the MESSAGE SYSTEM is DOWN.

Check if the Temporary Messages Hold Files (MSHDR, MSTXT, MSNOTIFY) are exist and in which library they were found. IF they do exist and were found in the ELSYSOVR library, then pass 'TF' to STATUS CODE.

**@Notes:**

This program was created and stewarded by the Information System Support (ISS) Department's ISA-Applications Group.



**ARMS Process Report**

This program is currently executed by the following programs prior to the execution of the Send e-Message program:

ARMS \*PSSR Program Error Paging & Messaging (AMPSSR) program  
ARMS Application Edit/Router (AM0040V1) batch program  
ARMS Non-Inbound Format Data Replacement Alteration (AM0047V1) program  
ARMS Internal/External Error Handling (AM0098) program

The program is executed with one output parameter value of a 2 character STATUS CODE that has one of the following values:

'OK' - Message System is UP (available) and DDM Connections Tested.  
'TF' - Message System is DOWN (unavailable), but the Temporary Messages Hold Files are in place to receive messages that are being requested to be sent.  
'ER' - Either Message System is down and the Temporary Messages Hold Files are NOT in place to receive messages that are being requested to be sent or DDM Connection FAILED.

@Files: (CRUD) (All are checked for existence and which in library they were found.)

MSHDR - Message System Message to be Sent - Header/Abstract  
MSTXT - Message System Message to be Sent - Detail Text Lines  
MSNOTIFY - Message System Message to be Sent - Users/Distribution

**Process**

Hierarchical numeric ID: 1.1.1.1.3.17  
Coded name: SNDEMSG  
Name: PGM Send Enterprise Message (SNDEMSG)  
Comment: @Definition: This is the executable ILE RPG command for sending enterprise message

@Purpose: To send previously outputted "e-message" to the specified ARMS On-Call Message Distribution List or the ARMS Technical Management from an another program.

**@Operational Method:**

Using the QTEMP library temporary work files as input for the formatted message, read the Message Program Send Msg to employees file to obtain the message distribution.

**@Notes:**

This program was created and stewarded by the Information System Support (ISS) Department's ISA-Applications Group.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.18  
Coded name: AM0098  
Name: PGM Handle Internal/External Error (AM0098)  
Comment: @Definition: This is the newer ARMS Internal/External Error Handling Program. This program performs functions beyond both the AM0090V1 and AM0097V1 program.

### ARMS Process Report

**@Purpose:** To notify the ARMS Application Development Department On-Call personnel of a transaction error condition that requires research and possible manual intervention. Error (ER) group type transaction data set in response to a rejected transaction data set received from a Rental Management Trading Partner host application system and to log the error to an error log file.

(NOTE: This program will replace AM0090V1 and AM0097V1 with the new error handling process. Error records are written to the error files AMERRLOG and ANDQER and transaction files ANDIST or AMPACK depending on what machine the program is currently running on. The program currently contains both the old and new error handling logic.)

**@Operational Method:**

- Verifies each non-blank passed input parameter error reason code that is not equal to '99' or '00' exists in the AMERRTBL Valid Error Codes table file and returns an 'A' for accepted or an 'R' for rejected. If the passed error reason code does not exist in the AMERRTBL file, AMPSSR program is executed to notify the ARMS On-Call staff of the program-to-program referential discrepancy and the erroneous error code value is concatenated with the constant text "ERROR CODE IS INVALID" and this resulting value is written to the AMKFLOG file record's FILLER field and its ERROR CODE field is loaded with the value '99' (Unknown).

- If the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to 'AMPSSR' passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

**@Notes:** This program is only executed by other ARMS never-ending programs or called programs that cannot process a ARMS transaction processing request as being in error or not doable and rejected it from further processing. This program is currently called with parameters by the following programs: AM0040V1, AM0046V1, AM0060V1, AM0061V1. (NOTE: This program is yet to replace the call to AM0097V1 older Internal Error Handler program in many other programs: AM0025V1, AM0030V1, AM0047V1, AM0062V1, AM1080V1, AM2090V1, CC2090V1 and EC2090V1.) The parameters used in calling this program are the following:

Input	5 character	COMPANY ID
Input	9/0 numeric	TRANSMISSION CONTROL ID#
Input	5/0 numeric	GROUP CONTROL ID#
Input	2 character	GROUP TYPE CODE
Input	20 character	CUSTOMER TRANSACTION ID
Input	20 character	VENDOR TRANSACTION ID
Input	6 character	FORMAT ID IN ERROR
Input	40 character	ERROR REASON CODE(S) (2 characters per
code/20 occurrences maximum)		
Output	1 character	PROGRAM RETURN CODE (Possible values:
'R'=Rejected - default, or 'A'=Accepted)		
Input	10 character	PROGRAM NAME IN ERROR (Rejecting the
transaction processing)		
Input	150 character	Transaction key value
Input	128 character	Transaction data value
Input	10 character	Program ID that is rejecting transaction
Input	4 character	Enhanced Error Type Code
Input	10 character	Data Queue Name that provided rejected
transaction key as input.		
Input	1 character	Error Severity flag

ARMS Process Report		
---------------------	--	--

Input	1 character	Error Transaction Type flag (possible values: 'O'=Outbound, 'I'=Inbound, and 'T'=Transfer) (currently not referenced by this program)
-------	-------------	---

Input	8 character	Alpha Add Date (ccyyymmdd format)
Input	6 character	Alpha Add Time (24-hour format)
Input	10 character	Calling Program ID
Input	5 character	Add Employee ID

@Files: (CRUD)

- AMPACK (C---)
- ANDIST (C---)
- ANDQER (CRU-)
- AMPERT (-R--)
- AMERRTBL (-R--)
- AMERRLOG (C---)

@Constants:

"ERROR CODE IS INVALID" as suffix for invalid error code.

'RTVMSGSTS' is the called program to check if the Message System Status is active.

'ARMS TEST MESSAGE' is the Test Message Header Text that is not referenced within the program.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.19

Coded name: AM2050V1

Name: PGM Retrieve Machine ID (AM2050V1)

Comment: @Purpose: To return the associated MACHINE ID (host platform name) for a given rental location.

@Operational Method:

- This program attempts to retrieve a branch information record from the Office Directories Branch Office (OFFDRB) file for the passed GROUP ID and BRANCH ID specified as input in the parameter list.

- IF both the passed GROUP ID and BRANCH ID are blank, then return '\*\*\*ERROR' as MACHINE ID to the calling program.

- IF the passed BRANCH ID is blank, then attempt to retrieve the first record associated for the passed GROUP ID.

..IF found, then return its MACHINE ID value.

IF the passed BRANCH ID is not blank, then attempt to retrieve the record associated for the passed GROUP ID and BRANCH ID values.

..IF branch is found, then return its MACHINE ID value for that branch office location record to the calling program.

@Notes:

Executed with the following parameters:

Input	2 character	GROUP ID
Input	2 character	BRANCH ID
Output	8 character	MACHINE ID

<b>ARMS Process Report</b>
----------------------------

**@Embedded Data/Constants:**

'\*\*\*ERROR' is the constant value loaded if no OFFDRB file record is retrieved successfully.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.20

Coded name: AM2051V1

Name: PGM Check for Holiday / After Hours (AM2051V1)

Comment: @Definition: This program determines if a given rental location is not able to accept new ARMS authorized reservations and detours it to Claims Connection .

@Purpose: To return the associated MACHINE ID (host platform name) for a given rental location's passed GROUP ID and BRANCH ID.

**@Operational Method:**

- Receives three (3) input parameters:

2 character GROUP ID

2 character BRANCH ID

1 character ARMS Profile ID "Send to Claims After Hours?" Flag

- Returns a single output parameter:

8 character MACHINE ID (NAME)

- The following decision table/grid shows which location a new or transferred authorized reservation will be routed based on this flag setting and other criteria:

**Active****Rental**

Location		Send to Claims After hours	Holiday	Weekend	Claims	Branch	Routed Open	Open
Claims		Y		Y	N/A	N/A	N/A	
N/A	Claims	Y		N	Y	N/A		
Y	ECARS	Y		N	N	Y		
N	Claims	Y		N	N	Y		
Y	ECARS	Y		N	N	N		
N	ECARS	Y		N	N	N		
		N		N/A	N/A	N/A	N/A	
	ECARS							

Retrieve the current system date, day of week and time (adjusted to that location's time zone).

**ARMS Process Report**

IF the passed input parameter value for the ARMS Profile flag "Send to Claims After Hours?" is "N" (No), then do the processing to determine if the passed GROUP ID and BRANCH ID is an active daily rental branch location.

Attempt to re-retrieve the associated record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field in the Office Directory Branch Master (OFFDRB) file by the passed GROUP ID if the passed BRANCH ID is blank, or use GROUP ID and the BRANCH ID.

Also, IF a record exists for just the passed GROUP ID, then indicate that the branch supports daily rental business, ELSE use the retrieved OFFDRB file record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field value.

Additionally, IF an OFFDRB file record was retrieved successfully AND its STATUS CODE is "A" (Active) and its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is "Y" (Yes), then load the MACHINE ID output parameter field with blanks. IF today is NOT a national holiday, then,

IF the associated OFFDRB file record was not successfully retrieved, OR the retrieve record's STATUS CODE is NOT "A" (Active), OR its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is NOT "Y" (Yes), then load the returned MACHINE ID output parameter field with '\*\*\*ERROR'.

ELSE, load the MACHINE ID field with blanks

IF the passed location is not found, or not a daily rental business location, or its status is not active, then return "\*\*\*ERROR" in the MACHINE ID parameter.

IF the passed input parameter value for the ARMS Profile flag "Send to Claims After Hours?" is "Y" (Yes), then do the processing to determine if today is a national holiday:

- Attempt to retrieve the associated Office Directory Branch Master (OFFDRB) file record using the passed GROUP ID and BRANCH ID.
- IF unsuccessful, then indicate that today is a holiday.
- IF successful, then using the retrieved OFFDRB file's MACHINE ID value access the associated ISS: Global Attributes (GACTL) File record to retrieve this branch's platform's associated COUNTRY NAME. Then use today's system date and this retrieved COUNTRY NAME value to attempt to access an existing record in the ECR: National Holiday (RA005P00) file. IF a record exists, then indicate that today is a national holiday.

Attempt to re-retrieve the associated record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field in the Office Directory Branch Master (OFFDRB) file by the passed GROUP ID if the passed BRANCH ID is blank, or use GROUP ID and the BRANCH ID.

Also, IF a record exists for just the passed GROUP ID, then indicate that the branch supports daily rental business, ELSE use the retrieved OFFDRB file record's BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION field value.

Additionally, IF an OFFDRB file record was retrieved successfully AND its STATUS CODE is "A" (Active) and its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is "Y" (Yes), then load the MACHINE ID output parameter field with blanks. IF today is NOT a national holiday, then,

IF the associated OFFDRB file record was not successfully retrieved, OR the retrieve record's STATUS CODE is NOT "A" (Active), OR its BRANCH BUSINESS TYPE FOR DAILY RENTAL INDICATION is NOT "Y" (Yes), then load the returned MACHINE ID output parameter field with '\*\*\*ERROR'.

ELSE, load the MACHINE ID field with blanks.

### ARMS Process Report

IF today is NOT a national holiday, then check the date to determine if its day of week is Saturday or Sunday.

IF not found, then Also, check if the today is a national holiday by checking for the existence of a record in the ECR: National Holiday (RA005P00) file by today's system date and the Country Name from retrieving the ISS: Global Attributes (GACTL) File by the MACHINE NAME from the retrieved Office Directory Branch Master (OFFDRB) file. This is to determine if the office is open and will accept new business.

IF it is a holiday, it is Saturday or Sunday, or the branch office is closed at the branch's current local time, then attempt to retrieve the Office Directory Branch File record for GROUP ID = "76" and BRANCH ID = "80" and check if this Claims Connection location is currently open with the current time adjusted to its local time zone. IF it is open, then return this GROUP ID and BRANCH ID, along with its current MACHINE ID value. IF it is not open, then return the original passed rental location's MACHINE ID for the previously retrieved closed branch office location record.

The ARMSPR1 Send to Claims After Hours flag (P1YN18) to determine whether to send a new reservation to Claims Connection or the closest Group/Branch. If this flag is set to 'N' for a trading partner, the new reservation will not be sent to Claims Connection whether it is a holiday or not. This will prevent authorizations for Canadian locations being sent to Claims Connections on U.S. national holidays.

@Files: (CRUD)

OFFDRB (-R--) Office Directories Branch Office File, keyed by GROUP ID and BRANCH ID.

GACTL (-R--) ISS: Global Attributes File

RA005P (-R--) ECR: National Holiday File

(Example: '19980701')

'CANADA' is the Canadian national holiday for their Independence Day.)

@Notes:

Executed with the following 4 input parameters:

Input	2 character	GROUP ID
Input	2 character	BRANCH ID
Input/Output	8 character	MACHINE ID (distributed application database host platform logical identifier/identifying name)
Input	1 character	SEND TO CLAIMS CONNECTION AFTER HOURS INDICATION FLAG

Currently only executed from the ARMS Application Edit/Router Batch (AM0040V1) program.

@Embedded Data/Constants:

'\*\*\*ERROR' is the constant value loaded if no active daily rental business type Office Directory Branch Location (OFFDRB) file record is retrieved successfully.

'Mon', 'Tue', 'Wed', 'Thu', 'Fri' are used to associate the correct OFFDRB file record's Open Time and Close Time for each Day of the Week.

'Sat' and 'Sun' are constant values used to determine if the current day of the week is on a weekend.

ARMS Process Report
---------------------

76 is the constant overriding value for the Claims Connection GROUP ID.  
80 is the constant overriding value for the Claims Connection BRANCH ID.

**@Improvement Opportunities:**

Only attempt to retrieve the associated OFFDRB file record in a single routine. Currently done in both the \$OFFDR subroutine to Retrieve OFFDRB (Office Directory File) Record and the \$HOLDY subroutine to Check RACNTR to determine if Holiday.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.21

Coded name: AM2053V1

Name: PGM Retrieve State Code (AM2053V1)

Comment: @Definition: This is the callable OPM RPG program to Retrieve State Code for a given Rental Location ID.

@Purpose: To extract and return the associated State Code from the Daily Rental Office Location Directory file for a given Rental Location ID.

**@Operational Method:**

Program accepts a two parameter list, composed of an input field, LOCATION ID (10 character), and an output field, STATE CODE (2 character). A record is attempted to be retrieved from Daily Rental Office Directory (DROPLF1) file with Location ID as its access key.

IF a record is found, the State Code will be extracted from the retrieved record's ADDRESS LINE 2 (CITY, STATE, ZIP) field value and will be loaded to the output parameter STATE CODE to be returned to the calling program. ELSE, set the STATE CODE to '\*\*\*' and return to the calling program.

@Notes: This callable program is executed by the following ARMS programs:

ARMS Application Edit/Router Batch (AM0040V1)

ARMS Add Surcharge (AM0096V1)

ARMS Generate Underage Surcharge (AM1085V1)

@Files: (CRUD)

DROFF (-R--)

**@Embedded Data/Constants:**

'\*\*\*' is the returned output STATE CODE if record is not retrieved.

**@Improvement Opportunities:**

1. Convert from OPM RPG/400 to ILE RPG/400 service program that is over all Office Directory files (internal and external organizations' rental locations) that will provide also Office Name, Street Address #1, Telephone Number and other information so that the ARMS application is separated from the Office Directories database access software tier.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.22

Coded name: AM0047V1

Name: PGM Enforce rules of Data Exchange for Outbound Transaction (AM0047V1)

<b>ARMS Process Report</b>
----------------------------

**Comment:** @Purpose: To generically enforce the rules specified by the ARMS Rental Management Trading Partners for data exchange on transaction data sets sent to adjusters from Enterprise Rent-A-Car or Claims Connection. This can comprise replacing data in a transaction data set's record formats' data elements that were from the rental application database with values retrieved from the ARMS centralized database files' data that had been populated by the transactions received from the adjuster. This will then "echo" back information to the trading partner's application that had been sent to ARMS.

**@Operational Method:**

Using the passed TRADING PARTNER PROFILE ID and ARMS DATABASE FILE FIELD NAME, attempt to retrieve the associated ARMS Trading Partner File/Field Control Data (ARMSPR4) file.

IF record not found, return to calling program without loading the PROGRAM RETURN CODE parameter.

IF record is found, then based on the two flags in the trading partner profile, use the following decision table:

**CASE 1 : FORMAT field is blank DATABASE field is NOT blank.**

	Mandatory To Send	Allow Field Changes	Action	
	Y		N/A	assign
database field				
format field error				to
code 04, missing				
corrected from				
database				
	N		N	assign
database field				to
format field if				
rental status not				
Waiting and not				
Unconfirmed				

**CASE 2 : BOTH fields are Non-Blank**

	Mandatory To Send	Allow Field Changes	Action	
	N/A		N	assign
database field				
format field if				to
rental status not				
Waiting and not				
Unconfirmed				

**CASE 3 : BOTH fields are Blank.**



ARMS Process Report			
Mandatory To Send	Allow Field Changes	Action	
Y		N/A	error
code 01, missing			
could not correct			
In all other cases, the format field remains unchanged regardless of the database field value.			
@Notes:			
This program is currently only called by the ARMS Database update and routing (AM0046V1) program with the following multiple parameters:			
Input	50 character	ARMS Database File field value	
Input/Output	50 character	Transaction Set Record Format field	
Old/Replaced value			
Input	6 character	ARMS Database File Field Name	
Input	1 character	ARMS Rental Cross-Reference File	
Record's STATUS CODE			
Input	5 character	COMPANY PROFILE ID	
Input	9/0 numeric	TRANSMISSION CONTROL ID	
Input	5/0 numeric	GROUP CONTROL ID	
Input	2 character	GROUP TYPE CODE	
Input	20 character	CUSTOMER TRANSACTION ID	
Input	20 character	VENDOR TRANSACTION ID	
Input	6 character	FORMAT ID IN ERROR	
Output	40 character	ERROR REASON CODES (20 occurrences maximum @ 2 characters, however, only first occurrence used.)	
Output	1 character	PROGRAM RETURN CODE (Not loaded by this program, but passed on from executed AM0097V1 program.)	
Input	10 character	CALLING PROGRAM ID IN ERROR	
@Files: (CRUD)			
ARMSPR4 (-R--) ARMS Trading Partner File/Field Control Data file			
FILE80 (C---) Send a Message 80-position work file (created empty and overridden to in the QTEMP library by the execution of 'CRTF80' program, then opened explicitly.)			
@Embedded Data/Constants:			
'RTVMSGSTS' is the constant defined in order to execute the program before calling the Send e-Message (MS0010) program.			
The following compile-time array "SBJ" is composed of 10 constant text lines (47 characters each) used for loading e-message header and detail lines for any mandatory data to be sent that is missing from the record format and the ARMS database:			
1.) MANDATORY TO SEND FIELD IS MISSING			
2.) Format:			
3.) Program:			
4.) Company:			
5.) Transmission Control ID:			
6.) Group Type:			
7.) Cust Tran ID:			
8.) Vndr Tran ID:			
9.) is missing-corrected from database			

**ARMS Process Report**

10.) is missing-could not be corrected from database

**@Improvement Opportunities:**

1.) Convert this OPM RPG/400 program to an ILE RPG module and have this data replacement enforcement logic used as the transaction data set's record format's field are being loaded.

2.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.23

Coded name: AM0096V1

Name: PGM Generate Pseudo-Authorization Change Transaction Data Set (Approved Govt.Surcharges) (AM0096V1)

Comment: @Purpose: To generate proprietary approved surcharge formats from the formats stored in the pre-approved surcharges file, AMSURTBL, based on the state and rental location.

**@Operational Method:**

- Delete all automatically generated (ORIGIN CODE = 'A') in Authorized Surcharge Detail file, AMSURD that are associated with the passed VENDOR TRANSACTION ID and generate AMMNTLOG ARMS Database Maintenance Log file records via call to AM1050V1.

- Pass the CURRENT RENTAL LOCATION ID to the called program AM2053V1 to retrieve the STATE CODE.

- Search AMSURTBL file for any records with BLANK COMPANY PROFILE ID whose LOCATION ID field value matches with the passed CURRENT RENTAL LOCATION Group ID or Group/Branch ID value. For every blank Location ID record or matches the passed CURRENT RENTAL LOCATION ID read, write an AMSURD Approved Surcharge Detail File record for that retrieved AMSURTBL file record's SURCHARGE CODE, SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values.

- Search AMSURTBL file for any records with the passed COMPANY PROFILE ID (that was embedded within the passed Transaction Data Set Key) whose LOCATION ID field value matches with the passed CURRENT RENTAL LOCATION ID Group ID or Group/Branch ID value. For every blank Location ID record or matches the passed CURRENT RENTAL LOCATION ID read, write an AMSURD Approved Surcharge Detail File record for that retrieved AMSURTBL file record's SURCHARGE CODE, SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values.

- Using the passed CURRENT RENTAL LOCATION ID Group/Branch ID, retrieve the associated BC026P00 file record. For any non-zero miscellaneous government surcharges that are applicable for this rental location, write an AMSURD Approved Surcharge Detail File record for that SURCHARGE AMOUNT and SURCHARGE CHARGE FREQUENCY field values loading the SURCHARGE CODE with '05' (Government).

- Return to calling program by explicit SETON \*INLR.

@Notes: This program is called from AM0046V1 program when processing a transaction data set for the following conditions: an Authorization (AT) group type for an add, change, or transfer that contain a SURD01 record format; an opening Rental Notification (RN) group type; or, an Request for Authorization (RA) Change

### ARMS Process Report

group type where the Date of Birth or the Rental Location value has been changed from a previous value. This called program is completed and removed from the job's program stack completely when \*INLR is SETON.

@Files: (CRUD)

- AMSURTB ( -R-- )  
 - BC026P ( -R-- )  
 - OFFDRB ( -R-- )  
 - AMSURD (CR-D)  
 - RAGBCHBT ( -R-- ) \* TEXAS GOVT SURCHARGE TO PRINT FOR 'I'nsurance  
 TYPE CUSTOMERS

@Constants:

'RECORD DELETED' is the message text for a value passed to AM1050V1 Add Maintenance Log Record program.

@Improvement Opportunities:

- 1.) If this program remained resident in the job's program stack, the overhead in opening all of the files could be reduced by making the file opens user-controlled to only occur if they are closed.
- 2.) Convert this OPM RPG/400 program to an ILE RPG module and have this data surcharge automatic generation logic executed as the transaction data set's record format's fields are being interfaced with the rental system database on the distributed rental application system host platform.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.25  
 Coded name: AMRVTSQ  
 Name: PGM Assign Next Vendor Transaction Id (AMRVTSQ)  
 Comment: @Definition: This is the ARMS Assign Next Vendor Transaction ID OPM RPG callable program.

@Purpose: To construct the next unique Vendor Transaction Identifier value for use in creating a new ARMS Rental Transaction Cross-reference File record.

@Operational Method:

Attempt to retrieve the LAST USED VENDOR TRANSACTION ID from the data area AMSEQCTL and lock it for update.

IF there was an error accessing this data area, load VENDOR TRANSACTION ID with 'ERROR' and return to the calling program.

IF there was no error in accessing and locking this data area, then increment its IDENTIFICATION NUMBER value by 1 and concatenate this 9 digit integer to the passed input 1 character MACHINE ID value and load the first 10 of 20 characters of a candidate VENDOR TRANSACTION ID.

Then, for every IDENTIFICATION NUMBER that is non-zero, ensure that an ARMS Rental Cross-Reference File record does not exist for the this derived Vendor Transaction ID by execution of the ARMS Cross-Reference Inquiry (AM2010V1) program in a loop until a return code of 'E' (No Cross-Reference Found) is sent, continuing to increment by 1 the numeric portion of this candidate VENDOR TRANSACTION ID.

### ARMS Process Report

Update the data area with the last derived Vendor Transaction ID value when the RETURN CODE of 'E' was returned to this calling program along with the Last Changed Date, Time and Program ID.

Return the new Vendor Transaction ID to the calling program.

@Notes:

This program is executed by following ARMS never-ending programs (NEPs):

ARMS Database update and routing (AM0046V1)  
 ARMS Dispatch Outbound Transactions (AM0060V1)  
 ARMS Outbound transaction editor (AM0062V1) (For non-ARMS Rental Management Trading Partners' Electronic Invoicing.)

It is executed with the following parameters:

Input	1 character	MACHINE ID
Input	10 character	PROGRAM ID
	Output 10 character	VENDOR TRANSACTION ID (to be used for cross-reference file record creation)

@Embedded Data/Constants:

'ERROR'  
 'E'  
 'AM2010V1'

@Improvement Opportunities:

Convert this OPM RPG/400 program to an ILE RPG service program.

Process

Hierarchical numeric ID: 1.1.1.1.3.26

Coded name: AM1099

Name: PGM Retrieve Next Transmission Control Id Number (AM1099)

Comment: @Purpose: To retrieve, generate and return the next ARMS internal Transmission Control ID number for use in transmissions from the distributed machine to the centralized host or to another distributed machine as in a transfer. This number is unique on its originating machine but is not unique across the whole ARMS.

@Operational Method:

The program is called passing two parameter fields, the first input and the second is output:

Input	CALLING PROGRAM ID
Output	TRANSMISSION CONTROL ID

Attempt to retrieve the next internal TRANSMISSION CONTROL ID value from the ARMS Next Outbound/Transfer Internal Transmission Control Sequence Number (AMRNAKSQ) file's single record and lock it for update.

If there is a database error, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the current timestamp's day of month, hour, minute and second (in the format: Oddmmhhss, with a zero in the first leading position, followed by the 2-digit dd = day of month, followed by 6-digit current 24-hour time to the whole second.).

### ARMS Process Report

If the record does not exist, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the numeric value of 1, increment that number by 1 and write a single AMRNAKSQ file record.

If the record does exist, then check if the retrieved NEXT TRANS SEQUENCE is zero, then load the output parameter for TRANSMISSION CONTROL ID NUMBER with the numeric value of 1. Else, load the output parameter for TRANSMISSION CONTROL ID NUMBER with the current NEXT TRANS SEQUENCE field value. Increment that NEXT TRANS SEQUENCE number field value by 1. If the updated NEXT TRANS SEQUENCE number field value is now zero, then set that value to 1. Update the single AMRNAKSQ file record.

Return the INTERNAL TRANSMISSION CONTROL ID TO BE USED to the calling program.

@Files: (CRUD)

- AMRNAKSQ (CRU-)

@Notes: When a transmission is to be sent to the trading partner, it is placed in the packaging file, AMPACK from where the packaging program, AM0120 picks it up and retrieves the next external transmission control id from a data area to assign it to this transmission. This external transmission control id exists only on the centralized host as opposed to the internal id that exists on all computer platforms and is unique only by machine.

@Improvement Opportunities:

- The file should be changed to a data area since there will always be only one record in the file, similar to AMSEQCTL (and use the empty file as a layout only for external data structure purposes). This may require an exclusive lock allocation (via an API or ILE RPG pointers) to wait less than 5 seconds, then reattempt until successful locking for input and writing the updated data storage, then issuing a deallocate/unlock. Also, this program should be duplicated to an RPG-ILE service program once some of the calling programs are converted to RPG-ILE programs. Additionally, CCARMSV1 no longer calls this program, therefore remove this dependent code. Lastly, this program is missing any program exception handling, like a \*PSSR subroutine calling AMPSSR. However, it may not be needed.

- 'CCARMSV1' used to check if is the value of the CALLING PROGRAM ID input parameter, then SETON \*INLR since this program is being called by an interactive program. (CCARMSV1 no longer calls this program, this processing requirement was moved to CC00INV1, therefore remove this code.)

#### Process

Hierarchical numeric ID: 1.1.1.1.3.27

Coded name: AM1050V1

Name: PGM Create Sensitive Fields Change Log (AM1050V1)

Comment: @Purpose: To generate an ARMS Rental Transaction Database Files' Changes and Deletes (Maintenance) audit log file record with the passed input Vendor Transaction ID, File Prefix Code, Field Name suffix, the previous "before-change", its current "after-change" and the Functional Group Type values for ARMS database files' field value changes. This is used for auditing or troubleshooting purposes.

@Operational Method:

This program will accept a parameter list composed of most of the data fields in the AMMTLOG record, excluding the Record Added Date and Time audit fields.

**ARMS Process Report**

(There are no record change stamp fields since this is a history of database changes or deletes log file.) The parameters (all input) are:

20 character	Vendor Transaction ID
2 character	File Code-File prefix
4 character	Field Code-Field suffix
50 character	Data - New value
50 character	Data - Previous value
2 character	Group Type Code
10 character	Calling Program ID
5 character	Employee ID

This program will accept input parameters provided and add a record to ARMS Rental Transaction Database Files' Changes and Deletes (Maintenance) audit log (AMMNTLOG) file. The record add stamp fields are loaded along with the passed input parameters values.

**@Files:**

AMMNTLOG (C---) Keyed by Vendor Transaction ID

**@Improvement Opportunities:**

- If any ARMS Database file fields increase in length beyond 50 characters, then the previous and current data value fields in AMMNTLOG and this parameter list (along with the programs that execute this program) will need to match this increase and the existing data will need to be converted to the changed format files. Likewise, if any of the ARMS Database file field names increase beyond 6 characters or the first 2 characters no longer denote file prefix, then these parameters, programs using this program and the AMFLTBL and AMFLDTBL fields will need to be changed also.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.28

Coded name: AM1080V1

Name: PGM Generate Transaction Credits (AM1080V1)

Comment:.. @Purpose: To generate transaction credit log records and to accumulate the TOTAL CUMULATIVE REVENUE and TOTAL CUMULATIVE TRANSACTION COUNT for a trading partner.

**@Operational Method:**

1) Attempt to retrieve the associated Transaction Cross-Reference file, AMXREF, record by the passed Vendor Transaction ID to retrieve its Company Profile ID, Customer Transaction ID, Rental Location ID and Ticket ID. If unsuccessful retrieval, execute AM0097V1 program with passed Error Code equal to '98' and return to calling program.

2) Determine if an associated Transaction Volume Discount Profile (ARMSPR7) exists for this trading partner. If the associated profile record does not exist, then return to calling program.

3) If the last 6 characters of Location ID is blank, then the first 4 characters contain an ECARS Group/Branch ID. Therefore, use the Company Profile ID to attempt to retrieve the associated ARMSPR5 ARMS Company Profile ECARS Rental System Variables file record. Else, use the Company Profile ID to attempt to retrieve the associated ARMSPR6 ARMS Company Profile Claims Connection Rental System Variables file record. If an associated Company Profile file record was not

### ARMS Process Report

retrieved, execute AM0097V1 program with passed Error Code equal to '67' and return to caller program.

4) If the passed Processing Code is 'P'ayment Advice, then perform detail calculations to compute the total amount billed to the trading partner.

5) If the passed Processing Code is 'P' or 'R', then perform detail calculations to compute the TRANSACTION CREDIT AMOUNT.

6.) Increment the retrieved ARMSPR5 or ARMSPR6 file record's PAYMENT ADVICE ACCUMULATED PAYMENT TOTAL COUNT (this is essentially the cumulative transaction count) and ACCUMULATED PAYMENT TOTAL AMOUNT (this is essentially the cumulative revenue).

8) Write record to AMTRNCR (Month-to-Date transaction credit file) after loading the following fields, including current date/time stamp fields:

- Populate the INSURANCE TOTAL RATE/DAY with the BILLED DAILY TOTAL AMOUNT (calculated above).

- Load the TRANSACTION CREDIT AMOUNT from the retrieved ARMSPR7 file record's DISCOUNT AMOUNT A field.

- IF the passed Processing Code is 'P', then load the TOTAL TICKET AMOUNT and the AMOUNT BILLED CUSTOMER from their associated fields from the retrieved AMIEBT file record's fields' values. ELSE (Processing Code = 'R' - Closing RN), clear the fields to zeros.

- IF the passed Processing Code is 'P', then load the INSURANCE TOTAL RATE/DAY field with the previously computed INSURANCE MINIMUM TOTAL AMOUNT value. ELSE (Processing Code = 'R' - Closing RN), clear the field to zeros.

- If the retrieved ARMSPR7 file record's DISCOUNT AMOUNT B field value is not zero, then do the following to calculate the ESTIMATED UTILIZATION FEE: If the retrieved ARMSPR7 record's DISCOUNT FREQUENCY B value is 'P' (Percent), then multiply the DISCOUNT AMOUNT B by the decimal ratio of the DISCOUNT FREQUENCY B field value. If the DISCOUNT FREQUENCY B value is 'L' (Lump) then use this value.

- Load the BILLED DATE from the retrieved AMIEBD record's ADD DATE.

- Load the following date subfields with the current system date: APPROVAL CENTURY/YEAR, APPROVAL MONTH, APPROVAL DAY.

@Complex Calculations:

BILLED DAILY TOTAL AMOUNT calculation:

1.) Attempt to retrieve the associated AMAUTD Authorization Detail file record for this Vendor Transaction ID to obtain the AUTHORIZED DAILY RATE, NUMBER OF DAYS AUTHORIZED, MAXIMUM DAILY RATE COVERED, POLICY MAXIMUM COVERED, BILL-TO PERCENT. If unsuccessful retrieval, execute AM0097V1 program with passed Error Code equal to '68' and return to caller program.

2.) If the NUMBER OF DAYS AUTHORIZED is greater than zero, then retrieve each associated AMIEBD Invoice Detail file record that has an ITEM CODE equal to 05 (Daily Rate Detail).

a.) For each record daily charge invoice detail record read,

i.) Save the least of ITEM RATE (Rental Rate Charged), AUTHORIZED DAILY RATE, MAXIMUM DAILY RATE COVERED as the RATE AMT.      ii.) Save the lesser of ITEM QUANTITY (Days Charged) or NUMBER OF DAYS AUTHORIZED as the DAYS QUANTITY.

ARMS Process Report
---------------------

iii.) If saved DAYS QUANTITY is greater than zero, then calculate  
 $BILLED\ AMOUNT = RATE\ AMT * DAYS\ QUANTITY.$

Subtract DAYS QUANTITY from NUMBER OF DAYS AUTHORIZED.  
 Accumulate BILLED DAILY TOTAL AMOUNT by BILLED AMOUNT.

c.) Multiply BILLED DAILY TOTAL AMOUNT by AUTHORIZED BILL-TO PERCENT (AMAUTD) to give a new BILLED DAILY TOTAL AMOUNT value.

d.) If the POLICY MAXIMUM COVERED amount is not zero, compare it the BILLED DAILY TOTAL AMOUNT. Add the lesser value to the appropriate retrieved ARMSPR5 / ARMSPR6 file record's ACCUMULATED PAYMENT TOTAL AMOUNT value giving a new ACCUMULATED PAYMENT TOTAL AMOUNT field value.

#### TRANSACTION CREDIT AMOUNT calculation.

1.) Retrieve (without update) the other rental system's Company Profile File (ARMSPR5 or ARMSPR6) for this Profile ID. Add together the new ACCUMULATED PAYMENT TOTAL COUNT fields from the two files (ARMSPR5 and ARMSPR6) with the increment of 1 for this transaction, giving a TOTAL ACCUM. COUNT (If not found, then set Error Code equal to '67').

2.) Use the current Company Profile ID and TOTAL ACCUM. COUNT result from (1) to SETTL on ARMSPR7 and retrieve the next record to get the amount of the transaction credit.

3.) If ECARS invoice, then retrieve the Group/Branch's associated Region Code from the OFFDRB Office Directory Branch File associated Group/Branch record. Else, it is a Claims Connection invoice, then leave region blank and use '7680' as Group/Branch ID.

4.) Retrieve the associated AMIEBT Invoice Total File record to get the Rental Ticket Total Amount and the Amount Billed to the Direct Bill Customer (error code 68 if not found) .

#### @Notes:

This program is only executed as a called program from the AM0046V1 never-ending batch program in the ARMS subsystem only on the ARMS application centralized host computer system platform whenever it processes the first closing 'RN' Rental Notification transaction data set for a non-zero Bill-To Percent authorized rental whose profile indicates auto-generation of a (PM) (currently as GEICO using ARMS/400), or, whenever it processes a 'PM' Payment Advice transaction data set.

Executed with the following passed input parameters:

9/0 packed numeric	Transmission Control ID Number
5/0 packed numeric	Transmission Group Control ID Number
20 character	Vendor Transaction ID
1 character	Functional Group Transaction Type Processing

Code ('P'-PM or 'R'-Closing RN).

@Files: (CRUD)

ARMSPR7	(-R--)
AMAUTD	(-R--)
AMIEBD	(-R--)



ARMS Process Report
---------------------

AMIEBT	(-R--)
OFFDRB	(-R--)
ARMSPR5	(-RU-)
ARMSPR6	(-RU-)
AMTRNCR	(C---)

**@Embedded Data/Constants:**

'7680' is loaded as default non-ECARS location in the Group/Branch ID data structure field to load the Group ID and Branch ID fields in the written AMTRNCR file record.

PGMNAM and PPRPGM are both set to 'AM1080V1'. (Suggest using the Program Data Structure subfield for Program ID in future.)

**@Improvement Opportunities:**

1.) Convert this OPM RPG program to an ILE RPG module that is part of the current AM0046V1 program.

2.) Since this program is attempting to do the same calculations that was done at Close a Rental Contract calculation time, I suggest that the computation of the Daily Rental Direct Billing Sub-Total Amount on each rental contract be stored in a new Rental Authorization Detail file's associated Sub-Total Amount for Daily Rental, along with Approved Surcharges Charged Sub-Total Amount and Taxes Charged Sub-Total Amount. Then, if the centralized ARMS database needs to retain this information for transaction credit purposes, a new internal format, IEBT02, can be loaded during "Generate Electronic Billing Invoice" in the transaction set and can be utilized and stripped off from the outbound transaction set that is sent to the trading partner. However, it would be simple if these new fields were interrogated from a centralized archive repository of closed rental contracts for the ARMS Transaction Credit Reports run monthly that primarily use the AMTRNCR file.

3.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

4.) Delete the constant 'AM1080V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.

5.) Convert program from OPM RPG to ILE RPG.

6.) Replace the use of the AM2010V1 program with the AM1010V1 program called for "I"nquiry.

**Process**

Hierarchical numeric ID: 1.1.1.1.3.29

Coded name: AM1010V1

Name: PGM Inquire/Update Cross - Reference (AM1010V1)

Comment: @Purpose: To retrieve for inquiry or update an existing associated ARMS Rental Transaction Cross-Reference File record when passed with the appropriate parameters.

**@Operational Method:**

- The following four input combinations can be passed to this program:  
 ARMS VENDOR TRANSACTION ID;  
 ARMS TRADING PARTNER COMP.PROFILE ID and CUSTOMER TRANSACTION ID;

**ARMS Process Report**

RENTAL LOCATION and the RENTAL CONTRACT/TICKET ID; or  
RENTAL LOCATION and the RENTAL BRANCH RESERVATION ID.

Likewise, records may be updated or added to the ARMS Rental Cross-Reference File by executing this program in "Maintenance" mode. Only inquiry is allowed if executed in "Inquiry" mode.

1.) The RENTAL LOCATION is always required and either a non-blank TICKET ID or RESERVATION ID must be passed.

IF RENTAL LOCATION, TICKET ID and RESERVATION ID are blanks, then load the RETURN CODE with 'E' (Error).

2.) IF Rental Location and Reservation ID are passed with the program call, use AMXREFL2 to retrieve the requested ARMS Cross-Reference File record.

a.) IF Ticket number is also passed with the program call, retrieve the record using AMXREFL2. Then validate the Ticket number passed with the program call against the Ticket data field XATKNO.

b.) IF passed TICKET ID and retrieved TICKET ID are different, then load the RETURN CODE with 'T' (Ticket ID Mismatch) and return to the calling program.

c.) IF the passed TICKET ID is same as the TICKET ID found for the reservation in the ARMS Cross-Reference File, then update all fields in the parameter list from AMXREFL2, set the RETURN CODE to blanks (blanks indicate successful) and return to the calling program.

3.) IF non-blank RENTAL LOCATION and TICKET ID are passed with the program, use AMXREFL1 to retrieve the data.

IF no record found for RENTAL LOCATION and TICKET ID, then load the RETURN CODE with 'E' (Error).

4.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with XACUID in AMXREFLn.

IF they are different, change the RETURN CODE to 'P' for 'Profile Error' and return to calling program.

5.) IF non-blank ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, retrieve record from AMXREFL3.

a.) IF TICKET ID / RESERVATION ID is passed, compare these with values retrieved from AMXREFL3.

IF they do not agree, change return code to 'T' / 'R'.

b.) IF RENTAL LOCATION is passed, compare this value with value retrieved from AMXREFL3.

IF they do not match, change return code to 'L'.

6.) IF an ARMS VENDOR TRANSACTION ID is passed, retrieve record from AMXREF.

a.) IF an ARMS COMPANY PROFILE ID is passed with the program call and a record is retrieved, compare this value with the associated field in the retrieved ARMS Cross-Reference File.

**ARMS Process Report**

IF they are different, change the return code to 'P' for 'Profile Error' and return to calling program.

b.) IF RENTAL LOCATION is passed, compare this value with the AMXREF value. IF they are different, change return code to 'L'.

c.) IF TICKET ID or RESERVATION ID is passed, compare these values with values retrieved from AMXREF.

IF they do not agree, change RETURN CODE to 'T' / 'R'.

7.) IF a record is retrieved from AMXREF or any of these logical files based upon the file use hierarchy, and additional data fields are passed, and more than one data mismatch occurs, change the return code to 'M' for multiple errors.

Hierarchy for record retrieval:

- 1.) IF ARMS VENDOR TRANSACTION ID is passed, and any other data fields, retrieve from AMXREF.
- 2.) IF ARMS COMPANY PROFILE ID and CUSTOMER TRANSACTION ID are passed, and any other data fields (except VENDOR TRANSACTION ID), retrieve from AMXREFL3.
- 3.) IF RENTAL LOCATION and RESERVATION ID are passed, retrieve from AMXREFL2.
- 4.) IF RENTAL LOCATION and TICKET ID are passed, retrieve from AMXREFL1.

RECORD MAINTENANCE FUNCTIONS:

1.) ACTION CODE passed is 'M' for maintenance. It is required to pass the VENDOR TRANSACTION ID for all maintenance transactions. Additional fields are optional, but only the physical file AMXREF is used for maintenance. ADD/CHANGE PROGRAM ID should also be sent.

2.) IF any existing AMXREF record is being maintained, the data fields are updated with the passed values along with record change stamp fields.

3.) IF new AMXREF record is being written, the data fields are updated with the passed values along with record add stamp fields.

4.) IF an AMXREF record is retrieved, compare the contents of each data field in the record with parameter containing that field.

IF they are different, create a maintenance log for the change. This is done only if parameter value is not blanks. Any non-blank value contained in the parameter list is logged and mapped to the AMXREF. AM1050V1 is called to create AMMNTLOG record.

RETURN CODE possible values:

blank: Successful retrieval / edit  
L: Location passed is blank / invalid  
#: Ticket/Reservation passed is blank  
T: Ticket passed, unmatched on AMXREF record  
P: Profile passed, unmatched on AMXREF record  
E: No AMXREF record found  
M: Multiple errors

ARMS Process Report
---------------------

ACTION CODE possible values:

-----  
 I: Inquiry  
 M: Maintenance

- IF the passed CALLING PROGRAM ID is NOT 'AM0046V1', then execute call to the ARMS Program Exception Handler 'AMPSSR' program, passing the CALLING PROGRAM ID and the Transaction's 96-character key value that is comprised of the passed TRANSMISSION CONTROL ID#, GROUP CONTROL ID#, GROUP TYPE CODE, CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID, padded with blanks.

@Notes:

This program is only executed by other ARMS never-ending programs that are processing a ARMS transaction request to inquire or update the ARMS Cross-Reference File.as being in error or not doable and rejected it from further processing. This program is currently called with parameters by the following programs:

ARMS Database Update and Routing program (AM0046V1)  
 ARMS Dispatch Rental Systems Request program (AM0060V1)  
 ARMS Distributed Edit and Dispatch Inbound Transactions program  
 (AM0061V1)  
 ARMS Distributed Edit of Generated Transaction Sets program (AM0062V1)

The parameters are internally defined in this program and programs executing this program are the following parameter fields:

Output	1 character	RETURN CODE
Input/Output	20 character	VENDOR TRANSACTION ID
Input/Output	5 character	TRADING PARTNER CO. PROFILE ID
Input/Output	20 character	CUSTOMER TRANSACTION ID
Input/Output	10 character	RENTAL LOCATION
Input/Output	6 character	RESERVATION ID
Input/Output	6 character	TICKET ID
Output	1 character	STATUS CODE
Output	8/0 numeric	STATUS CODE LAST CHANGED DATE
Output	6/0 numeric	STATUS CODE LAST CHANGED TIME
Output	10 character	STATUS CODE LAST CHANGED PGM ID
Output	1 character	MACHINE ID
Output	10 character	BRANCH CLAIMS OFFICE CODE
Output	1 character	RENTAL SYSTEM SOURCE ID
Output	8/0 numeric	RECORD ADD DATE
Output	6/0 numeric	RECORD ADD TIME
Output	10 character	RECORD ADD PROGRAM ID
Output	8/0 numeric	RECORD CHANGE DATE
Output	6/0 numeric	RECORD CHANGE TIME
Output	10 character	RECORD CHANGE PROGRAM ID
Input	2 character	GROUP TYPE CODE
Input	1 character	ACTION CODE

@Files: (CRUD)

- AMXREF (CRU-) AMXREFL1, L2, L3

@Embedded Data: (Constants)

'AM1010V1 ' is used as the program's name.

### ARMS Process Report

The following values are named constants used for sending the ARMS Cross-Reference File (AMXREF) record's field name suffix for the passing of any field value modified as a FIELD CHANGED parameter field value in the call to the ARMS Transaction Database Maintenance Logging (AM1050V1) program.

'CUID'  
'CUTI'  
'RNLC'  
'RENO'  
'TKNO'  
'STCD'  
'STDT'  
'STTM'  
'STPG'  
'MCID'  
'SOID'  
'TEID'  
'ADDT'  
'ADTM'  
'ADPG'  
'CGDT'  
'CGTM'  
'CGPG'

Also, there is a compile-time array to store the values for the formatted OS/400 commands:

CLOF OPNID(AMXREF )  
OPNDBF FILE(AMXREF ) OPTION(\*ALL)

#### @Improvement Opportunities:

Converting this OPM RPG program to ILE RPG service program. It might negate the need for the OS/400 command execution OPNDBF and CLOF of the AMXREF file.

Replace the 'AM1010V1 ' constant with the PROGRAM ID field of the Program Status Data Structure.

Replace the internally defined parameter fields in this program and all programs executing this program with a database file with no record to be used as an externally defined data structure so that only one definition of the parameter's data elements exist, along with comments on their purpose and possible values.

This program can replace the call to the ARMS Cross-Reference File Inquiry (AM2010V1) program in many other programs if it is called in "Inquiry" mode.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.30

Coded name: RTVMCHA

Name: PGM Retrieve Machine Emulation Attributes (RTVMCHA)

Comment: @Purpose: To retrieve the emulated system (machine) platform attributes important for Enterprise programs that have been standardized for a machine.

This program is executed when executing the user-written command of the same name, "RTVMCHA" (Retrieve Emulated Machine Attributes).

#### @Operational Method:

To avoid adversely impacting performance, all of these steps are executed only the first time a job executes the RTVMCHA program. Subsequent execution in

### ARMS Process Report

production will bypass the program initialization steps, only returning previously loaded values.

Upon program initialization, do the following:

- executes the program to retrieve the system (machine) name (RTVSYs)
- opens GACTL and MACHID files
- retrieves the system name's records from both GACTL and MACHID
- moves data from both GACTL and MACHID into its parameter list
- closes GACTL and MACHID files
- execute the /Q Check DEV's machine without emulation program (ERTVDEV) to determine IF actually on "DEV". IF true, return the previously derived values to the calling program and end program and return to caller.

After program initialization has been performed and this inactive program was re-executed by the same job's calling program, only do the remaining processing step, return the previously derived values to the calling program.

@Files (CRUD)

MACHIDENPF (-R--) Machine Information File, via keyed access path  
 MACHID#1 by key element MACHINE ID (8 character)  
 GACTL (-R--) Global Attributes file, by key element MACHINE ID (8 character)

@Notes:

This program is a Internal Systems Support Group utility tool software callable program, executed with a single parameter defined in an external data structure in its physical file (\$\$EMACH). All its output parameter subfield data elements, their attributes and a description can be found in source text member ELLIB/QDOCSRC/RTVEMCHA.

This program is especially important for each of our software applications needs the flexibility to accommodate the multi-national differences we will encounter in day-to-day business dealings since R-R-A-C is now operating outside the North American continent. For example, what we consider a standard method of displaying the date (MM/DD/YY) is not the standard in the United Kingdom; they expect to see it displayed as DD/MM/YY.

To facilitate expansion of business operations into the United Kingdom, and eventually into other European countries, a Global Attributes file (GACTL) was created to contain information that varies from country to country. By using the system name obtained through RTVSYs, a program can access the GACTL file and use that information to tailor display screens and reports to its needs. For example, the GACTL file contains a field to tell programs to display the date as MM/DD/YY or DD/MM/YY.

Process

Hierarchical numeric ID: 1.1.1.1.3.31  
 Coded name: AM1085V1  
 Name: PGM Generate Underage Driver Surcharge (AM1085V1)  
 Comment: @Purpose: To delete any preexisting and write any needed ARMS Pre-Approved Authorized Surcharge Detail file (AMSURD) records with the CHARGE TYPE CODE = '20' (Underage Driver Surcharge) and the CHARGE ORIGIN CODE = 'U' (Underaged Driver Surcharge Generation program).

@Operational Method:

Confidential

Page 66 of 246

8/11/00

**ARMS Process Report**

1. Call the Retrieve ARMS Cross-Reference File Record (AM2010V1) program using the passed VENDOR TRANSACTION ID field value and its required parameters to retrieve the associated ARMS Profile ID value.

2. IF the returned SOURCE ID = 'B' (Branch Reservations), then call the Retrieve ARMS ECARS-Specific Profile File Record (AM2020V1) program to retrieve from the associated ARMS ECARS-Specific Profile File (ARMSPR5) record its flag field value for automatically generating underage driver approved surcharge authorizations (INSURANCE WILL PAY UNDERAGE='Y').

ELSE (SOURCE ID = 'C' - Claims Connection), call the Retrieve ARMS ECARS-Specific Profile File Record (AM2030V1) program to retrieve the same information from the associated ARMS-Claims Connection Specific Profile file (ARMSPR6) record.

3. IF this retrieved profile's flag field value is 'Y' (Yes), then do the following:

a. Using the passed VENDOR TRANSACTION ID, retrieve the associated Authorization Detail file (AMAUTD) record. IF successful, derive (compute) the DRIVER'S AGE by subtracting the DATE OF BIRTH from the RENTAL START DATE.

b. Call the Retrieve Rental Location's State Code Inquiry (AM2053V1) program, passing the called Retrieve ARMS Cross-Reference File Record (AM2010V1) program's returned parameter value for RENTAL LOCATION ID to retrieve the RENTAL LOCATION'S STATE CODE value.

c. Using the retrieved PROFILE ID value and RENTAL LOCATION'S STATE CODE value, retrieve the associated ARMS Profile Control Fields by State Code file (ARMSPR8) record.

IF unsuccessful, reattempt using the value '\*' in place of the RENTAL LOCATION'S STATE CODE value. If still unsuccessful, execute return processing below.

d. IF the derived DRIVER'S AGE is greater than or equal to the the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE THRESHOLD AGE value, then use the passed VENDOR TRANSACTION ID value and SURCHARGE TYPE CODE = '20' (Underage Driver) to retrieve and delete any associated AMSURD file record. For each record deleted, the program will write a record to the ARMS Maintenance Log file by the calling of Generate ARMS Maintenance Log Records (AM1050V1) program with loaded parameter values.

e. IF the derived DRIVER'S AGE is less than the the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE THRESHOLD AGE value and the INSURANCE WILL PAY UNDERAGE is equal to 'Y', then write an associated ARMS Surcharge Detail file (AMSURD) record, using the passed VENDOR TRANSACTION ID value, SURCHARGE TYPE CODE = '20' (Underage Driver Surcharge Authorization Type Code), the retrieved ARMS Profile Control Fields by State Code file (ARMSPR8) record's UNDERAGE CHARGE AMOUNT value with a CHARGE FREQUENCY set to 'D' (Daily Charge) and the CHARGE ORIGIN CODE = 'U' (Underaged Driver Surcharge Generation program).

@Notes:

ARMSPR8 is currently empty.

This program is called from the AM0046V1 ARMS update never-ending program executing in a batch mode on the centralized ARMS application host computer system platform when processing a transaction data set for the following conditions: an Authorization (AT) group type for an add, change, or transfer that contain a AUTD01

### ARMS Process Report

record format; an opening Rental Notification (RN) group type; or, an Request for Authorization (RA) Change group type where the Date of Birth or the Rental Location value has been changed from a previous value. This called program is completed and removed from the job's program stack completely when \*INLR is SETON and RETURN is executed.

IMPORTANT: ARMS Profile Control Fields by State Code File (ARMSPR8) is currently empty in the production environment. Therefore, this program is called, loaded and initialized, then explicitly is ends, removed from the program stack and control is returned to the calling program.

@Files: (CRUD)

- AMAUTD (-R--) ARMS Authorization Detail File
- ARMSPR8 (-R--) ARMS Profile Control Fields by State Code File
- AMSURD (CR-D) ARMS Approved Surcharge Detail File

#### Process

Hierarchical numeric ID: 1.1.1.1.3.32

Coded name: AM0052

Name: PGM Determine Nearest Location by Phone Number (U.S. Only) (AM0052)

Comment: @Purpose: To determine the closest United States rental branch to a specified telephone number's area code and exchange using the Horizontal and Vertical Wire Coordinates locator database.

@Operational Method:

At the start, file VERHOR gives the vertical and horizontal coordinates of a specified area code and exchange number.

These coordinates are given in "units". There is 316 units per mile.

This program scans a 30 mile area around the VERHOR coordinates.

This is accomplished by calculating 4 new coordinates.

2 Horizontal and 2 Vertical.

The specified area code and exchange generates coordinates X2 and Y2 from the VERHOR file.

To "draw" the 30 mile box we subtract 9,480 units (30 miles) from X2 and Y2 to get X1 and Y1.

We then add 9,480 units to X2 and Y2 to get X3 and Y3.

Using the X1 and Y1 coordinate, attempt to retrieve the closest next record in the Daily Rental Offices (DROFF) file. This file contains the name, address...etc. of all daily vehicle rental office locations. Along with this information, the vertical and horizontal coordinates of each office are found in this file. By comparing these coordinates to the "box" coordinates we decide whether or not this office falls within the specified 30 mile area as read in as an input variable from the Search Input Parameters for Locator System (RANGE) data area.

The number of miles from the originally specified area code/exchange to each rental office is arrived at by calculating the square root of the combined horizontal and vertical coordinates of each rental office.

VWIRE = the vertical coordinate of the rental office.

HWIRE = the horizontal coordinate of the rental office

DVWIRE = the vertical coordinate of the original Area Code/Exchange

DHWIRE = the horizontal coordinate of the original Area Code/Exchange



### ARMS Process Report

When a location is found that is within boundaries, check the "Prevent ARMS Reservations" flag. IF this flag is 'N', include the location. If this flag is 'Y' and there is no "Forward To Location" (DROAGB) field value, skip this location. IF this flag is 'Y' and there is a forwarding location, save the first forwarding location. IF no other Enterprise location is found, use the saved forwarding location.

All office information is stored in array "AC", which is sorted in ascending sequence according to mileage. The closest Enterprise office is used if one exists.

#### @Notes:

This program is currently executed only by the ARMS Detail Set Edits and Assign Location (AM0040V1) OPM RPG program called with the following parameters:

Input	3 character	AREA CODE	(comprised of non-leading zero integers only)
Input	3 character	EXCHANGE	(comprised of non-leading zero integers only)
Input	1 character	COMMISSION CODE	
Output	10 character	RENTAL LOCATION CODE	
Output	1 character	MACHINE ID	
Output	1 character	SOURCE ID	

#### @Files: (CRUD)

OFFDRGRPPF (-R--) OFFDG#2  
 VERHOR (-R--)  
 DROFF (-R--) DROFLF1, DROFFL2

#### @Embedded Data/Constants:

30 miles is the maximum radius/distance for closest Enterprise rental branch to pickup a renter.

#### @Improvement Opportunities:

Change the calling program to execute the AM0053 program instead for all trading partners but specifically profiled ARMS Trading Partner Company.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.33  
 Coded name: AM0053  
 Name: PGM Determine Nearest Location by Longitude / Latitude using Phone Number or Postal Code (Canadian Only) (AM0053)  
 Comment: @Purpose: To locate offices in a specific area based on the telephone number provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any Canadian Enterprise offices.

#### @Operational Method:

-This program has four main logic sections that:

1) Draws a variable size box around a centerpoint of a phone number. Program is passed phone number and a 30 mile box is drawn around this number. Program calls AT\_002 and passes the

### ARMS Process Report

phone# and size in the parm. AT\_002 chains to the phone# file, gets the basepoint coordinates, and sends back via the parms the basepoint coordinates for that phone#, and the upper right and lower left longitude/latitude coordinates of the box, and a return code to indicate any errors. AT\_002 adjusts and corrects the box size for the curvature of the earth and is accurate to within 400 ft on Ave.

- 2) Retrieves all the records from the branch office file that fall within the box.
- 3) Calculates the distance (in miles), from the phone number centerpoint, of all the records retrieved.

The program calls 'AT\_001' program and passes a parameter that has the centerpoint (basepoint) and the longitude and latitude points of all records previously retrieved, and gets returned in the parameter the distances in miles for each set of points, and a PROGRAM RETURN CODE for errors. This is currently limited to 30 sets of points on each execution. IF more than 30 records are needed, a second(or third, fourth, etc.) execution can be made until all distances are calculated. This program scans a 30 mile area around the retrieved longitude and latitude coordinates and returns the .

- 4.) Find the rental location needed for this transaction. IF one does not exist, then send transaction to Claims Connection.

Make use of "Prevent ARMS Reservations" flag field (DROAMF) of the DROFF file. This field indicates whether unsolicited ARMS authorizations are to be sent to a group/branch. IF this field is 'Y', then send the authorization to the forwarding group/branch (if not blank) from the DROFF record. IF this flag is 'N', include the location. IF this flag is 'Y' and there is no "Forward To Location" field (DROAGB) value, skip this location. IF this flag is 'Y' and there is a forwarding location, save the first forwarding location. IF no other Enterprise location is found, use the saved forwarding location.

#### @Notes:

This program is currently executed only by the ARMS Detail Set Edits and Assign Location (AM0040V1) OPM RPG program called with the following parameters:

Input	10 character	TELEPHONE NUMBER
Output	10 character	RENTAL LOCATION CODE
Output	1 character	MACHINE ID
Output	1 character	SOURCE ID

It executes the following programs:

AT\_002 - Draw box around a centerpoint.

AT\_001 - Calculate Distance between points.

\*\* Will notify the application department when problem occurs executing programs.

#### @Files: (CRUD)

OFFDRGRPPF (-R--) Office Directories Group Master file, accessed by the OFFDG#2 access path key element, GROUP ID (2 character)  
 DROFF (-R--) Daily Vehicle Rental Office Locations file, accessed by the DROFFL7 access path key element, LONGITUDE and LATITUDE (both 9/6 numeric)

#### @Embedded Data/Constants:

### ARMS Process Report

30 miles is the maximum radius/distance for closest Enterprise rental branch to pickup a renter.

'7680' is the default rental location for Claims Connection.

'C' is the default MACHINE ID host platform (System ID='CENTRAL') and SOURCE ID (Claims Connection) values.

'ARMS' is the constant TERMINAL ID value for input parameter to the execution of the 'AT\_001' program.

#### @Improvement Opportunities:

Change the calling program (AM0040V1) to execute this program for all location once all rental locations have been loaded into the Longitude and Latitude Locator by Telephone Number database for all trading partners but specifically profiled ARMS Trading Partner Companies.

#### Process

Hierarchical numeric ID: 1.1.1.1.3.34

Coded name: NUS017A

Name: PGM Determine Nearest Enterprise Location by Phone or Postal Code (NUS017A)

Comment: @Purpose: To locate 1 to 9 offices in a specific area based on the telephone number or postal code provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any offices from the DROFF (NR : Daily Rental & Non-Enterprise Office Loc's) file, only Enterprise offices, or only Enterprise airport offices.

@Operational Method: The following parameters must be passed to the program with appropriate values set based on task at hand.

#### Input Params:

Location Phone#  
 Location Postal Code  
 Open Only Flag  
 Maximum Allowable Distance  
 Airport Branches Only Flag  
 Department ID  
 Accepting Reservations Flag  
 Retrn Forward-to Locatn Flag  
 Requested Date/Time  
 Time Zone Group/Branch ID  
 Opening Time Allowance  
 Closing Time Allowance  
 Return Code  
 Branches to Return 1-9

#### Output Params:

Location ID Array - these are the locations that satisfy the input search criterion.

Mileage Array - these are the corresponding distances for the locations from the reference point.

#### @Notes:

1) A testing stub called LOCTRUT exists in library ELARMS to test the program.

2) In order to receive only Enterprise offices, the calling program must do the following overrides :

ARMS Process Report
---------------------

Overriden file	Override to file
Droffl2	Droffl3
Droffl7	Droffl9

In order to exclude Ford locations, the calling program must do the following overrides.

Overriden file	Override to file
Droffl2	Droffl4
Droffl7	Droffl10

@Files	(CRUD)
DROFF	(-R--)

## Process

Hierarchical numeric ID: 1.1.1.1.3.35

Coded name: AM2020V1

Name: PGM Retrieve ARMSPR5 Information (AM2020V1)

Comment: @Purpose: To access ARMSPR5 (ECARS Profile) and ARMSPR1 (Trading Partner General Profile) record and return the details to the calling program.

## @Operational Method:

Executed with the following entry parameters and their usage:

Output 1 alpha	Program Return Code
Input 5 alpha	Trading Partner Profile ID
Output 10 alpha	ECARS Closing Calculations to Compute the
Direct Bill-To Total Due	
Output 11,2 numeric	Payment Total Amount
Output 1 alpha	Underaged Drivers Allowed (Y/N) indication
flag	
Output 1 alpha	Insurance pays underage driver indication
flag	
Output 1 alpha	Trading Partner has capability to recieve
Electronic Billing indication flag	

@Files: (CRUD)

ARMSPR1 (-R--)

ARMSPR5 (-R--)

## @Improvement Opportunity:

This program and AM2030V1 could be combined into a single program, especially if the ARMS Trading Partner Profile database files are further normalized.

## Process

Hierarchical numeric ID: 1.1.1.1.3.36

Coded name: AM2030V1

Name: PGM Retrieve ARMSPR5 Information (AM2030V1)

### ARMS Process Report

Comment: @Purpose: To access ARMSPR6 (Claims Connection Profile) and ARMSPR1 (Trading Partner General Profile) record and return the details to the calling program.

@Operational Method:

Executed with the following entry parameters and their usage:

	Output 1 alpha	Program Return Code
	Input 5 alpha	Trading Partner Profile ID
	Output 10 alpha	Claims Connection Closing Calculations to
Compute the	Direct Bill-To Total Due	
	Output 11,2 numeric	Payment Total Amount
flag	Output 1 alpha	Underaged Drivers Allowed (Y/N) indication
flag	Output 1 alpha	Insurance pays underage driver indication
Electronic Billing indication flag	Output 1 alpha	Trading Partner has capability to recieve

@Files: (CRUD)

ARMSPR1 (-R--)

ARMSPR6 (-R--)

@Improvement Opportunity:

This program and AM2020V1 could be combined into a single program, especially if the ARMS Trading Partner Profile database files are further normalized.

Process

Hierarchical numeric ID: 1.1.1.1.3.37

Coded name: RAS013A

Name: PGM Determine Nearest Enterprise Location by Phone or Postal Code (RAS013A)

Comment: @Purpose: To locate 1 to 9 offices in a specific area based on the telephone number or postal code provided and return the results and distance from point of inquiry to the requesting program. The request can be made for any offices from the DROFF (NR : Daily Rental & Non-Enterprise Office Loc's) file, only Enterprise offices, or only Enterprise airport offices.

@Operational Method: The following parameters must be passed to the program with appropriate values set based on task at hand.

Input Params:

Location Phone#  
 Location Postal Code  
 Country Code  
 Open Only Flag  
 Maximum Allowable Distance  
 Airport Branches Only Flag  
 Department ID  
 Accepting Reservations Flag  
 Retrn Forward-to Locatn Flag  
 Requested Date/Time  
 Time Zone Group/Branch ID  
 Opening Time Allowance

<b>ARMS Process Report</b>
----------------------------

Closing Time Allowance  
Return Code  
Branches to Return 1-9

Output Params:

Location ID Array - these are the locations that satisfy the input search criterion.

Mileage Array - these are the corresponding distances for the locations from the reference point.

@Notes:

1) A testing stub called LOCTRUT exists in library ELARMS to test the program.

2) In order to receive only Enterprise offices, the calling program must do the following overrides :

Overriden file	Override to file
Droffl7	Droffl9

In order to exclude Ford locations, the calling program must do the following overrides.

Overriden file	Override to file
Droffl7	Droffl10

@Files	(CRUD)
DROFF	(-R--)

### Process

Hierarchical numeric ID: 1.1.1.1.3.38

Coded name: NUS018A

Name: PGM Verify Branch is Open (NUS018A)

Comment: @Purpose: To send a GPBR and have the open/closed status returned. The request time can be the current date and time or a time in the future. The program uses the three Nat Res policy files, NRALPA2, NRPCDG, and NRGBM. It also uses GPBRTZ and a time zone reference field must be sent.

@Operational Method: The following paramters can be passed to the program:

Requested Group/Branch ID  
Requested Date/Time  
Time Zone Group/Branch ID  
Opening Time Allowance  
Closing Time Allowance  
Within Opening Time Allowance  
Within Closing Time Allowance  
Open Flag  
Return Code

The Open Flag and Return Code are the only return fields.

The program checks the Nat Res Policy Files to determine if the requested branch is open or closed.

Closing Time Allowance is available but not currently used by ARMS.

**ARMS Process Report**

@Files (CRUD)

NRALPA2 (-R-)  
NRPCDG (-R-)  
NRGBM (-R-)

**Process**

Hierarchical numeric ID: 1.1.1.1.3.39  
Coded name: DQAM55V1  
Name: DTQ Input for Program AM0055V1 (DQAM55V1)  
Comment: @Definition: DQAM55V1 is a data queue used to provide input to PGM AM0055V1 which maintains the files used for ARMS Online Reporting, accessed via ARMS/400 (green screen or gui).

**Process**

Hierarchical numeric ID: 1.1.1.1.3.40  
Coded name: AM0055V1  
Name: PGM Online Reporting File Write/Update Program (AM0055V1)  
Comment: @Purpose: To update and write records to the ARMS Group Type Summary File AM096P which keeps track of the number of times a group/type is transmitted for a customer profile and its vender transaction ID. Also to update and write records to the ARMS Online Reporting Detail file AM095P which keeps details on a rental keyed on customer profile and vender tran ID. To log transactions to ARMS Current days transaction for Sync program AM097P.

**@Operational Method:**

- Wait for entry(s) to exist in DTAQ (DQAM55V1).
- When a shutdown request is received (group type - SD), end program.
- When a non-shutdown request is received:
  - Create a record in AM096P01 to keep track of the number of transactions for a vendor tran id. If a record already exists add 1 to the total and update.
  - Check to see if a record exists in any of the following files by keying on Vender Tran ID: AMAUTD, AMRPRD, AMRNTD, AMSURD, AMIEBT, AMADJD, AMXREF
  - If a matching record is found move all of the appropriate data from AMXREF, AMAUTD, AMRPRD, AMADJD, AMRNTD, AMSURD, AMIEBT to the ARMS Online Reporting Detail File AM095P.
  - If there is not an existing record in AM095P00 one will be created, otherwise the existing record will be updated. All previous data in an existing record will be replaced with all data found in the files.
  - Check to see if a record exists in AM097P01. If record exists, skip record. If record is not found, write a record to the file.

@Files: (CRUD)

- AMXREF (-R--)  
- AMADJD (-R--)  
- AMAUTD (-R--)

<b>ARMS Process Report</b>
----------------------------

```

- AMIEBT      (-R--)
- AMRNTD      (-R--)
- AMRPRD      (-RU-)
- AMSURD      (CR-D)
- AM0095P00   (CRU-)
- AM009601    (CRU-)
- AM009701    (CR--)

```

@Embedded Data/Constants:

```

'YES'
'NO '

```

#### Process

Hierarchical numeric ID: 1.1.1.1.4

Coded name:

Name: AUT Send Transaction for Distributive Processing (AM100-AM101)

Comment: @Definition: The automatic process of sending the transaction from the centralized machine to the distributed machine for processing, via ICF files.

#### Process

Hierarchical numeric ID: 1.1.1.1.4.1

Coded name: AM0100

Name: PGM Distribute Routed Transaction from Centralized Host System (AM0100)

Comment: @Purpose: To send a received (inbound) or transferred ARMS transaction set from the centralized host platform to the distributed host platform.

@Operational Method:

Execution is started with a single MACHINE ID (1 character) input parameter field value.

For program initialization:

- Retrieve the current platform's CURRENT SYSTEM (MACHINE) NAME by execution of the Retrieve System Name (RTVSYS) program.

- Concatenate the Recieve Routed Ditributed Transaction on Distributed Host System Program Name, 'AM0101', with the passed input MACHINE ID value for deriving the specific Program Device Name.

- IF CURRENT SYSTEM NAME = 'DEV', change the ICF workstation device file EVOKE program's library to 'ELARMS', ELSE, make the EVOKE program's library 'ELLIB'

- Attempt to acquire this Program Device. Attempt 5 times before signalling an error. Introduce a delay of one minute between each attempt.

- IF the Program Device is acquired successfully, Evoke the derived ARMS Distribution Incoming Receive Program.

After program initialization has completed, for a repetitive cycle until a shutdown data queue entry is received, the following is performed:



### ARMS Process Report

- Receive the next keyed data queue entry from the specific keyed data queue for the key equal to the passed input MACHINE ID parameter value. Immediately from the DQAMAPH data queue until empty, then infinitely wait for DQAMAP1 data queue entries.

- Attempt to retrieve an associated record from the AMAPPS file using the data queue entry as the retrieval key value.

- IF at least one associated AMAPPS file record exists, then do the following:

- - Send the key to the ARMS Distribution Incoming Receive program.  
 - - Read all ARMS Set file records with the ARMS Application Data Queue Key.  
 - - Send each data record to the ARMS Distribution Incoming Receive program. When all records are read and sent, send a blank record.

@Files: (CRUD)

- AMAPPS (-R--)

@Embedded Data/Constants:

'AM0101' is the evoked program's specified name.

'ELARMS2' and '\*LIBL' for the evoked program's specified possible library names.

'DEV' to determine if being executed on the development (not production) software environment platform.

'\*DOWN' and 'SD' as the Profile ID and Group Type Code to indicate a data queue entry of a shutdown request.

'DQAMAPH' and 'DQAMAP1' are the Hold Distribution Input and the Transactions to Distribute Input data queue names.

'DLYJOB DLY(60)' is the OS/400 command executed to cause 1 minute delays between the 5 Attach attempts.

@Improvement Opportunities:

Convert this OPM RPG program to ILE RPG program.

... Possibly change this communication method from ICF to CPI-C.

#### Process

Hierarchical numeric ID: 1.1.1.1.4.2

Coded name: AM0101

Name: PGM Receive Routed Distributed Transaction on Distributed Host System (AM0101)

Comment: @Purpose: To receive routed transaction set from the centralized host platform.

@Operational Method:

- Acquire the program device 'AM0101'  
 - Receive routed transaction input from the centralized ARMS host platform.  
 - IF data was received  
   - Write records to the distributed Application Transaction File (AMAPP)  
   - Send data queue entry to Edit/Update/Distribute Transaction (DQAM61)

ELSE

ARMS Process Report
---------------------

- Send data queue entry to Edit/Update/Distribute Transaction (DQAM61)
- Release Device.

ENDIF

@Files: (CRUD)

- AMAPP (C---)

@Notes: The centralized ARMS host platform EVOKES the receive job on the distributed rental system host platform using an ICF connection. Presently, the distributed rental system host platform, excludes BIRELAND, VGERMANY, and UK.

#### Process

Hierarchical numeric ID: 1.1.1.1.4.6

Coded name: RTVSYS

Name: PGM Retrieve System Name (RTVSYS)

Comment: @Purpose: To retrieve the system name to pass back to another calling program.

@Operational Method:

Executed with the single output parameter, RETRIEVED SYSTEM NAME (8 character).

Retrieve the current system's network attribute of system name.

IF the RETRIEVED SYSTEM NAME = 'DEV', then execute the ISS DEV Machine Emulation (DEVMCHE) OPM CL program.

#### Process

Hierarchical numeric ID: 1.1.1.1.5

Coded name:

Name: AUT Interface with Rental Systems (AT/CN/EX/CM)

Comment: @Definition: The automatic process of dispatching a data queue entry for an appropriate program to update the Rental database be it either ECARS or CLAIMS CONNECTION.

#### Process

Hierarchical numeric ID: 1.1.1.1.5.1

Coded name:

Name: AUT Interface with ECARS (AT/CN/EX/CM)

Comment: @Definition: The automatic process by which the ECARS Rental database files are updated due to a request received from the Rental Management Trading Partner (AT, CN, EX, CM).

#### Process

Hierarchical numeric ID: 1.1.1.1.5.1.1

Coded name: EC00ATV1

Name: PGM Interface EC with Rental Authorization (EC00ATV1)

Comment: @Purpose: This is an ARMS-to-ECARS database update interface program that processes Rental Direct Billing Authorization 'AT' Functional Group Type for

**ARMS Process Report**

Adds or Changes sent by ARMS Insurance Customers or automatically generated authorization changes. This program will also process previously authorized reservation Rental Location's Group-to-Group 'AT' transfers.

**@Operational Method:**

This NEP receives the 'ATB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF the flag for Centralized Adjuster Phone number is 'Y' then retrieve from CUSTMAST the office phone number from that file record based upon customer number and use this phone number to populate ECARS data files, else the adjuster phone number supplied in the ADJD01 format will be used. This flag is retrieved via a 'B1' call to AM2090V1.

IF any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

IF while attempting to retrieve any existing records that are locked for update, the transaction is deferred by sending the data queue entry to a secondary input data queue (DQAM6BV1) to be reedited by the AM0061V1 program for later processing and a deferral notice data queue entry is sent to the primary input data queue (DQAM61V1) to notify that program that a deferred transaction exists (both data queue entries sent with key = 'DIS'). This will trigger the program AM0061V1 to start the wait timer so that when the wait is elapsed it can receive from the secondary data queue.

IF an authorization transfer is received, then a new ARMS authorized branch reservation is created, along with any applicable instructions, callback and prewrite information, such as date of birth.

IF an authorization add is received that has an invalid or no open rental contract/ticket or reservation identifier in it, then a new authorized rental reservation is created, along with any applicable instructions and prewrite information, such as date of birth.

IF an authorization add or change is received that has a valid open rental contract (ticket) and a reservation identifier in it, then update both the existing unauthorized open rental ticket and its "linked" branch reservation with the authorization information.

IF an authorization add is received that has a valid open rental contract (ticket) but no reservation identifier in it, then update the existing open rental ticket with the authorization information, a new authorized rental reservation is created, and this reservation is "linked" to the open contract. However, IF the open rental contract was opened using another non-ARMS authorized branch reservation or national reservation, then break the "link" of the previous open rental contract identifying information from the previous reservation. When a NatRes link is broken, a record is created in the file, NRDBTRAN, to appropriately credit NatRes.

### ARMS Process Report

In all cases that an existing branch reservation is updated with an authorization for the first time or an authorized branch reservation is created, a 'DIS' keyed data queue entry is sent to the DQAM60V1 data queue as input to the Dispatch Rental Systems Request (AM0060V1) program. This should cause an authorization confirmation (AC) group type transaction data sent to be dispatched for generation and sent.

In all cases that an existing open rental contract is updated with an authorization add or change, the associated adjuster callback direct billing authorization information is updated. For an authorization add only, the EXTENSION DATE is recomputed by adding this input transaction's data set's authorization detail (AUTD01) record format's AUTHORIZED DAYS to the open rental contract's START CHARGES DATE.

IF, however, the resulting EXTENSION DATE is a Saturday or Sunday, check the Trading Partner Profile's WEEKEND SLIDE FLAG. ..IF the WEEKEND SLIDE METHOD CODE is "Y"es and the EXTENSION DATE is Saturday or Sunday, then "slide" (change) the EXTENSION DATE to the date for the following Monday.

..IF the WEEKEND SLIDE METHOD CODE is "M"odified and the EXTENSION DATE is Sunday, then "slide" the EXTENSION DATE to the date for the following Monday.

Likewise for an authorization add only, callback detail notes are generated to indicate the recomputed EXTENSION with an "Authorized Until mm/dd/yy" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text where the "mm/dd/yy" is the computed new extension date and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format.

#### @Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

This program was changed to not remove the \*TX REMB miscellaneous government tax unconditionally on open contracts, but to instead check a flag in the Valid Business Types that the special TEXAS GOVERNMENT SURCHARGE is applied to (RAGBCHBT) file (via logical access path RAGBBT, by Business Type, Group ID and Branch ID) to see whether or not this tax should be charged.

#### @Files: (CRUD)

File	AMAPP	(-R--)	ARMS Application Interface Input Transaction
	RACPENDC	(-R--)	Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file
	RACBRNUM	(CRU-)	Branch Reservation ID Control File
	RACBRMST	(CRU-)	Branch Reservation Master File
	RACBRWRT	(CRU-)	Branch Reservation Prewrite File
Information File	RACINS	(CRU-)	Branch Reservation/Open Rental Contracts Insurance
	RACBRIST	(CRU-)	Branch Reservation Special Instruction
File	ECSURCHG	(CR-D)	ECARS Authorized for Rental Direct Billing Surcharges
	RACMAST	(-RU-)	Open Rental Contract/Ticket File
Information File	RACSMAS	(CRU-)	ECARS Open Rental Contract/Ticket Supplemental
	CB007P	(CRU-)	Rental Callback Control File, accessed by RACCBCTL logical view, by Group/Branch ID and Document ID
	RACCBDET	(CR--)	Rental Callback Detail Notes File
	RACCK	(-RU-)	Branch Reservation Credit Check File
	CUSTMAST	(-R--)	Customer Master File

ARMS Process Report
---------------------

RACCBKOC (-R--) Rental Consolidated Callback Control File  
 RACCBK3 (CRUD) Rental Consolidated Callback Control File  
 RACCB3 (CR-D) Rental Callback Consolidation Skeleton File - Shop  
 OFFDRB (-R--) Office Directories Internal Organization Branch  
 Master File  
 NR025P (-RU-) National Reservations Master File, accessed by  
 NR025L00 logical view, by National Reservation ID  
 NRDBTRAN (CRU-) National Reservations Database Transactions Update File

@Embedded Data/Constants:

'Until' used to provide a preformatted Callback Detail Notes phrase.  
 '/' used to as a date separator for date output into a Callback Detail  
 Notes text.  
 'ARMS CUST-' used to provide a preformatted RACBRIST File Sequence 001  
 record's comment text.

Process/shutdown codes:

'AT'  
 'SD'

Format IDs:

'APPD01'  
 'ADJD01'  
 'ADJD02'  
 'AUTD01'  
 'CMD01'  
 'INSD01'  
 'RNTD01'  
 'RNTD02'  
 'RPRD01'  
 'SURD01'  
 'RATD01'

Comments:

'VEHICLE NEEDED IMMEDIATELY'  
 'UNIT NEEDED AT OTHER#'  
 'UNIT NEEDED AT HOME#'  
 'UNIT NEEDED AT OFFICE#'  
 'INS CO WILL PAY '  
 '/DY' used as a Branch Reservation Rate Quoted Comment for the quoted  
 daily rental rate.  
 'Non numeric data received in Transmission/Group Control ID'  
 'LECO0ATV1' is used as the JOB NAME.  
 'ECO0ATV1' used as the CALLING PROGRAM ID.  
 'ATB' as the input key used for receiving keyed data queue entries.

Constants for formatting numeric fields to text:

'/DAY'  
 '/MAX'  
 '.,.'  
 '0123456789' all valid numeric characters to use for validation of all  
 numeric character data element fields.  
 'RATE + TAX'  
 ' + TAX'  
 'RATE +SCHG'  
 ' +SCHG'

### ARMS Process Report

The following OS/400 Override Printer file commands are executed through execution of the 'QCMDEXC' program to change the output queue names by insertion of the Group/Branch ID (4 character) in the output queue name to cause the printing of an open rental contract/ticket or the executed Print a Branch Reservation to print at the associated rental location's office plain paper printer:

```
OVRPRTF FILE(QPRINTP) OUTQ(P) FORMTYPE(PLAINPAPER)
DLTOVR FILE(QPRINTP)
```

The following OS/400 database file override commands are executed through execution of the 'QCMDEXC' program to decrease the locked record/object wait times for update files from the object default of 1 minute to 5 seconds:

```
OVRDBF FILE(RACBRNUM) WAITRCD(5)
OVRDBF FILE(RACBRMST) WAITRCD(5)
OVRDBF FILE(RACBRWRT) WAITRCD(5)
OVRDBF FILE(RACINS) WAITRCD(5)
OVRDBF FILE(RACBRIST) WAITRCD(5)
OVRDBF FILE(ECSURCHG) WAITRCD(5)
OVRDBF FILE(RACMAST) WAITRCD(5)
OVRDBF FILE(RACCBCTL) WAITRCD(5)
OVRDBF FILE(RACCBDET) WAITRCD(5)
OVRDBF FILE(RACSMAS) WAITRCD(5)
OVRDBF FILE(RACCCR) WAITRCD(5)
OVRDBF FILE(RACCBK3) WAITRCD(5)
OVRDBF FILE(RACCB3) WAITRCD(5)
OVRDBF FILE(NR025L00) WAITRCD(5)
OVRDBF FILE(NRDBTRAN) WAITRCD(5)
```

#### @Improvement Opportunities:

- 1.) Convert this OPM RPG program to an ILE RPG program.
- 2.) Delete the 'EC00ATV1' program name and 'LEC00ATV1' job name constants and specify/use the Program Status Data Structure's PROGRAM ID and JOB NAME data elements for these purposes.
- 3.) Move the "Authorized Until mm/dd/yy" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text comment templates into an external ARMS Table File that contains the preformatted callback detail note automated comments.
- 4.) Add a new executable program to check for an associated existing record in the file RACPENDC (CRU-) "Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file" for checking if OK to process and if so, output a record until the operation is complete, then delete the RACPENDC record that was written to clear way for other transactions.

#### Process

Hierarchical numeric ID: 1.1.1.1.5.1.2  
 Coded name: EC00EXV1  
 Name: PGM Interface EC with Rental Extension/Termination of Authorization (EC00EXV1)  
 Comment: @Purpose:

To update the rental database with the authorized extension days or termination date received from the trading partner.

#### @Operational Method:

**ARMS Process Report**

This program endlessly receives the 'EXB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF the any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

IF while attempting to retrieve any existing records that are locked for update, the transaction is deferred by sending the data queue entry to a secondary input data queue (DQAM6BV1) to be reedited by the AM0061V1 program for later processing and a deferral notice data queue entry is sent to the primary input data queue (DQAM61V1) to notify that program that a deferred transaction exists (both data queue entries sent with key = 'DIS').

IF the received transaction set's STATUS CODE is not "E"xtend nor "T"erminate, then reject the transaction set back to the trading partner's adjuster as a "data domain error".

In all cases that an existing open rental contract is updated with an authorization extension or termination, the associated open contract file record's and its adjuster callback file record's direct billing authorization information is updated if the adjuster information is changed.

IF this transaction is an authorization termination (CRED01:CRESTS='T'), then update the callback's current EXTENSION DATE is replaced with this input transaction's data set's customer request for extension detail (CRED01) record format's non-zero TERMINATION DATE, regardless if the new value of the EXTENSION DATE is a Saturday or Sunday date. Likewise, a callback detail note record is generated to indicate the replaced EXTENSION DATE with an " Rental Terminated On: mm/dd/yy" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text where the "mm/dd/yy" is the replaced EXTENSION DATE and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format. Update the number of authorized days in the Reservation record.

IF this is an authorization extension (CRED01:CRESTS='T'), with DAYS TO EXTEND is zero, then the current EXTENSION DATE remains. This will allow ARMS Trading Partner Insurance Company to send a termination request... (The second phase will allow ARMS Trading Partner Insurance Company to receive "RE" and send "EX" transactions.) Likewise, a callback detail note record is generated to indicate the same value EXTENSION DATE with an " Rental Terminated On: mm/dd/yy" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text where the "mm/dd/yy" is the current value of the callback's EXTENSION DATE and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format. Update the number of authorized days in the Reservation record.

IF this is an authorization non-zero extension, then the non-zero EXTENSION DATE is recomputed by adding this input transaction's data set's customer request for extension detail (CRED01) record format's DAYS TO EXTEND to the open rental contract's current EXTENSION DATE.

### ARMS Process Report

Likewise for an authorization extension for a non-zero DAYS TO EXTEND only, a callback detail note record is generated to indicate the recomputed EXTENSION with an "Rental Extended Until: mm/dd/yy" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text where the "mm/dd/yy" is the computed new extension date and the Adjuster's name is derived from the data provided in the Adjuster Detail (ADJD01) record format.

Update the number of authorized days in the Reservation record.

#### @Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

#### @Files: (CRUD)

File	AMAPP	(-R--)	ARMS Application Interface Input Transaction
	ECEXTCTL	(CRU-)	ECARS Pending Extension Request Control File
	RACMAST	(-RU-)	Open Rental Contract/Ticket File
	RACCBDET	(C---)	Rental Callback Detail Notes File
	RACCBDOC	(-R--)	Rental Consolidated Callback Control File
	RACCBK3	(CRU-)	Rental Consolidated Callback Control File
	CB007P00	(-RU-)	Open Rental Contracts Callbacks Control File
	CB032P00	(-RU-)	RMS Callback Contracts by Group/Branch/Ticket
	RACBRMST	(-RU-)	ECR Branch Reservations Master File

#### @Embedded Data/Constants:

Detail Notes	'Rental Extended Until:'	used to provide a preformatted Callback phrase.
Detail Notes	'Rental Terminated On:'	used to provide a preformatted Callback phrase.
Detail Notes	'Rental Extended 0 days'	used to provide a preformatted Callback phrase.
	'DQAM61V1'	
	'EC00EXV1'	
	'OVRDBF CB007P00 WAITRCD(*IMMED)'	
	'OVRDBF RACMAST WAITRCD(*IMMED)'	
	'OVRDBF RACMAST WAITRCD(*IMMED)'	

'/' used to as a date separator for date output into a Callback Detail Notes text.

#### Logicals:

'Y'  
'N'  
'0'  
'1'

#### Process/shutdown codes:

'EX'  
'SD'

#### Format IDs:

'APPD01'  
'ADJD01'  
'CRCD01'  
'CMD01'

#### Comments:



<b>ARMS Process Report</b>
----------------------------

'LEC00EXV1' is used as the JOB NAME.  
 'EC00EXV1' used as the CALLING PROGRAM ID.  
 'EXB' as the input key used for receiving keyed data queue entries.  
 '0123456789' as all valid numeric characters to use for validation of all numeric character data element fields.

The following OS/400 Override Printer file commands are executed through execution of the 'QCMDEXC' program to change the output queue names by insertion of the Group/Branch ID (4 character) in the output queue name to cause the printing of an open rental contract/ticket or the executed Print a Branch Reservation to print at the associated rental location's office plain paper printer:

```
OVRPRTF FILE(QPRINTP ) OUTQ(P      P) FORMTYPE(PLAINPAPER)
DLTOVR FILE(QPRINTP)
```

@Improvement Opportunities:

1.) Delete the 'EC00EXV1' program name and 'LEC00EXV1' job name constants and specify/use the Program Status Data Structure's PROGRAM ID and JOB NAME data elements for these purposes.

2.) Change the program's use of the database files used for update purposes before opened for update via the following OS/400 database file override commands are executed through execution of the 'QCMDEXC' program. This would be to decrease the locked record/object wait times for update files from the object default of 1 minute to 5 seconds.

3.) Add a new executable program to check for an associated existing record in the file RACPENDC (CRU-) "Open ECARS Rental Contracts/Tickets in progress of being changed or closed control file" for checking if OK to process and if so, output a record until the operation is complete, then delete the RACPENDC record that was written to clear way for other transactions.

4.) Move the "Rental Extended Until:", "Rental Terminated On:", "Rental Extended 0 days" and "By: AdjusterLastName\*AdjusterFirstName\*" formatted text comment templates into an external ARMS Table File that contains the preformatted callback detail note automated comments.

5.) Prior to opening the user-controlled open files that are used to READ or CHAIN for UPDATE, execute an "Override Data Base File" command to change the record-lock wait times from the default 60 seconds to 5 seconds as was done in the EC00ATV1 and EC00CNV1 programs.

Process

Hierarchical numeric ID: 1.1.1.1.5.1.3  
 Coded name: EC00CNV1  
 Name: PGM Interface EC with Rental Cancellation / Denial of Authorization (EC00CNV1)  
 Comment: @Purpose:

To update the rental database with the cancellation information received from the trading partner to either cancel a previously ARMS-authorized open contract and/or open branch reservation, or, to deny billing responsibility in response to a request for authorization on a branch reservation or an open rental contract (ticket).

@Operational Method:

**ARMS Process Report**

This program endlessly receives the 'CNB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF there are no deferred transactions while processing, then execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the current CALLING PROGRAM ID (10 character) to write the electronic customer messages to ECARS message database.

IF while attempting to retrieve for update any existing file records that are locked for update, then the transaction is deferred by sending the data queue entry with key = 'DIS' to a secondary input data queue (DQAM6BV1) and a deferral notice data queue entry with key = 'DIS' to a primary input data queue (DQAM61V1) to be reedited by the AM0061V1 program for later processing.

IF the received transaction set's reservation is not found, the reservation is closed, or closed ticket that had been opened using the ARMS-authorized reservation, then error the transaction and page ARMS On-call.

IF the received transaction set's reservation is closed, then execute the Internal/External Error Handling Program (AM0098) to Reject the transaction.

IF the received transaction set's retrieved unauthorized reservation has been voided, discontinue any further processing of this transaction.

IF the received transaction set's retrieved rental contract has been voided, then the transaction is deferred by sending the data queue entry with key = 'DIS' to a secondary input data queue (DQAM6BV1) and a deferral notice data queue entry with key = 'DIS' to a primary data queue (DQAM61V1) to be reedited by the AM0061V1 program for later processing. AM0061V1 will then handle the Ticket record not being found appropriately.

IF the received transactions set's retrieved reservation is authorized, then do the following:

..IF the received transaction set's reservation is found, is ARMS authorized, and, a ticket is currently being opened using the reservation (BRTKT = 999999), then defer this transaction as stated previously.

..IF the received transaction set's open ticket and/or reservation is found, is/are ARMS authorized, and available, then reset its direct billing indication (leave the Bill-To ID and related information), load the RATE QUOTED COMMENT with "ARMS Direct Bill Canceled By AdjusterLastName, AdjusterFirstName, and clear the ARMS authorization information from Branch Reservation File (RACBRMST) and from Ticket Master Files (RACMAST & RACSMASST) file records and its associated Callback Control and update. Create a Callback Control for the Reservation if it does not exist. Generate a Callback Detail Note file record with the comment "ARMS Direct Bill Canceled By AdjusterLastName, AdjusterFirstName" in NOTE #1, along with the associated adjuster's claim center internal CUSTOMER NAME from the Customer Master File (CUSTMAST) as "At customername" in NOTE #2. Update or create the consolidated callback control files. Callbacks for a reservation attached to an Incomplete Ticket will be created or updated using the Reservation number.

### ARMS Process Report

..IF the received transaction set's reservation is found, is NOT ARMS authorized, and is available for update (regardless if linked to open contract/ticket or not), then reset its (and its associated open rental contract/ticket) direct billing indication (leave the Bill-To ID and related information), load the reservation's RATE QUOTED COMMENT with "ARMS Direct Bill Denied By AdjusterLastName,AdjusterFirstName", in the Branch Reservation File (RACBRMST) and the Ticket Master File (RACMAST) file records and its associated open rental contract/ticket file and update. Update the callback and the consolidated callback control files, using the open rental contract/ticket identifier if it is completely open (has a non-zero CONTRACT DATE), else use the reservation's identifier. Create a Callback Control for the Reservation if it does not exist. Generate a Callback Detail Note file record with the comment "ARMS Direct Bill Denied By AdjusterLastName,AdjusterFirstName" in NOTE #1, along with the associated adjuster's claim center internal CUSTOMER NAME from the Customer Master File (CUSTMAST) as "At customername" in NOTE #2.

#### @Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

#### @Files: (CRUD)

AMAPP (-R--)  
 RACBRMST (-RU-)  
 CB007P00 (CRU-)  
 RACCBDET (C---)  
 RACMAST (-RU-)  
 RACCBLOC (-R--)  
 RACCBCTK (C-U-) ECR Callback Consolidation Skeleton File,  
     by Group ID, Branch ID, Rental Contract/Document ID via the RACCBK3  
 logical access path.  
 RACCBCBS (C-U-) ECR Callback Consolidation Skeleton File - Shop,  
     by Group ID, Branch ID, Rental Contract/Document ID via the RACCB3  
 logical access path.  
 CUSTMAST (-R--)  
 RACMAST (-RU-)  
 EMSG (C---)

#### @Embedded Data/Constants:

'ARMS'  
 'ARMS Direct Bill Denied By'  
 'ARMS Direct Bill Canceled By'

#### @Improvement Opportunities:

#### Process

Hierarchical numeric ID: 1.1.1.1.5.1.4  
 Coded name: EC00CMV1  
 Name: PGM Interface EC with Customer Message (EC00CMV1)  
 Comment: @Purpose:

To update the rental database with any text messages received from the trading partner in the functional group type 'CM'.

#### @Operational Method:

### ARMS Process Report

This program endlessly receives the 'CMB' keyed input data queue entries from the input data queue (DQAM61V1) that program AM0061V1 generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, read all of the associated ARMS Application Interface Input Transaction file (AMAPP) records.

IF any COMD01 or SURD01 record formats were found, execute the Update Electronic Messages program (ECMSGV1) with 3 input parameters, the APPD01 record format's Reservation ID (6 character), the data queue entry value (96 character), and the CALLING PROGRAM ID (10 character) to write the electronic messages to ECARS message database.

#### @Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters.

@Files: (CRUD)

AMAPP (-R--)  
RACBRMST (-R--)  
RACMAST (-R--)

#### @Embedded Data/Constants:

'EC00CMV1' used as the CALLING PROGRAM ID.  
'CMB' as the input key used for receiving keyed data queue entries.

#### @Improvement Opportunities:

- 1.) Convert this OPM RPG program to an ILE RPG program.
- 2.) Delete the 'EC00CMV1' program constant and specify/use the Program Status Data Structure's PROGRAM ID data element for this purpose.

#### Process

Hierarchical numeric ID: 1.1.1.1.5.1.6  
Coded name: AM2090V1  
Name: PGM Retrieve ARMS Data (AM2090V1)  
Comment: @Purpose:

To retrieve specified information from the ARMS database.

#### @Operational Method:

This program is executed using a single 256-byte data structure parameter for both input and output (see @Notes for a detailed description).

- Determine which type of information is being requested using the passed input parameter identification code.

-- Passed input parameter's data element for Data Structure ID Code specifies the associated information retrieval processing description:

**ARMS Process Report**

'B1' - Retrieves AMXBCO file record trading partner's branch claims office (BCO) information by the passed ENTERPRISE CUSTOMER ID

'B2' - Retrieves AMCLSTBL file ARMS Vehicle Class Description and Rate Code Table information by the passed ECARS VEHICLE CLASS CODE

'B3' - Retrieves AMXREF Rental Transaction Cross-Reference, AMXBCOL1 Branch Claims Office\*, and ARMSPR1 Trading Partner attributes information by the passed ENTERPRISE CUSTOMER ID, RENTAL LOCATION ID and RESERVATION//TICKET ID

'B4' - Retrieves AMXREF, AMXBCOL1\*, ARMSPR1, along with the Rental System Application Specific\*\* (ARMSPR5 or ARMSPR6) information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B5' - Retrieves AMXREF, ARMSPR1, along with the Rental System Application Specific\*\* (ARMSPR5 or ARMSPR6) information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B6' - Retrieves AMXREF, ARMSPR1 information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

'B7' - Retrieves AMXBCOL1, ARMSPR1 file information by the passed ENTERPRISE CUSTOMER ID

'B8' - Retrieves ARMSPR1 file record information by the passed ENTERPRISE CUSTOMER ID

'B9' - Retrieves AMXREF file record information by the passed RENTAL LOCATION ID and RESERVATION//TICKET ID

(\*Note: Branch Claims Office information is retrieved only when the retrieved Rental Transaction Cross-Reference indicates that it is not currently authorized or the Identification Code is 'B4'. This condition was needed since Trading Partner Profile ID was stored in two places. Undesirable results occurred prior to this condition.)

(\*\*Note: Rental Application System Profiles consists of two files: One for ECARS used by ERAC branches for the rental administration of their group's vehicles; and one for Claims Connection used by the National Reservations Call Center staff for the administration of third-party non-ERAC vehicle rentals. The Source ID on the Rental Transaction Cross-Reference ('B' = Branch ECARS; 'C' = Claims Connection) determines which file the information is retrieved.

- IF the passed Identification Code does not match those accepted by this program, generate a program exception error and return an unsuccessful Completion Status Code ('D') to the calling program.

- IF the requested information record(s) are successfully retrieved, return the values from those file(s) record(s) along with a successful Completion Status Code ('A') to the calling program.

- IF any record was not found, return blank value(s) from the specific file record along with a successful Completion Status Code ('A') to the calling program.

- IF a database error occurred, then generate a program exception error and return an unsuccessful Completion Status Code ('D') to the calling program.

- When the passed Identification Code is 'B4', determine if the trading partner's branch claims office is currently active for electronic billing.

<b>ARMS Process Report</b>
----------------------------

@Notes:

This program is currently executed with a single 256 character string input/output parameter that is further redefined using the empty ANDSB#V1 file as an external data structure. The external definition contains two important input fields. The first is the Data Structure Identification Code, which is always a two position alpha field in the first two positions of the data structure. The second immediately follows the first and is a one position Program Return Code. A value of 'A' returned in this position to the calling program indicates the processing request was completed successfully. The total maximum length is 256 positions, and the unused space has been defined as a "filler" (future expansion) field.

The data structure data elements descriptions are:

	Input	2 character	ARMS Identification Code
	Output	1 character	Program Return Code
	Input	7 character	Enterprise Customer ID
	Input	10 character	Rental Location ID
	Input	6 character	Reservation ID
	Input	6 character	Ticket ID
	Input	2 character	ECARS Vehicle Class Code
	Input	8,0 numeric	Transaction Date (in CCYYMMDD format)
	Input	10 character	Calling Program ID
	Output	20 character	Vendor Transaction ID
	Output	20 character	Customer Transaction ID
	Output	5 character	ARMS Profile ID
	Output	1 character	ARMS Rental Transaction Cross-
Reference File Record	Status Code		
	Output	1 character	Machine ID
Claims Connection	Output	1 character	Source ID - Branch Reservations /
	Output	10 character	Branch Claims Office - AMXBCO
	Output	14 character	Enterprise Car Class Code
	Output	20 character	ARMS Car Class Description
	Output	10 character	Billing Program
P1FG10	Output	1 character	Generate Extensions from Rental -
Rental - P1FG11	Output	1 character	Generate Batch Extensions from
	Output	1 character	Electronic Bill(Y/N) - P1YN12
	Output	1 character	Generate Policy Max Note P1YN17
- P1YN21	Output	1 character	View Ticket/Reservation 1 Day Early
Request - P1YN22	Output	1 character	Only Days Needed For New Callback
	Output	1 character	Electronic Messaging - P1FG24
- P1YN25	Output	1 character	Allow Cust Tran Id on Fax Documents
	Output	1 character	Allow Underaged Driver - P1YN26
	Output	1 character	Send ELCD01 Format on RA - P1YN27
XBYNBL	Output	1 character	Generate Electronic & Paper Bills -
	Output	1 character	Insurance Pays Underage Surcharge
N=No Slide - P1	Output	1 character	Weekend Slide Y=Sat/Sun M=Sun Only
	Output	1 character	R=RA I=IN B=Both N=No
	Output	1 character	How to Populate RATD01?
	Output	1 character	ARMS Special Condition
	Output	1 character	Send Adjuster Name on RA Y/N

ARMS Process Report			
	Output	1 character	Centralized Adjuster Phone #
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
	Output	1 character	Expansion Flag Field
Handling	Output	1 character	RC Cancel Assignment Surcharge
	Output	73 character	Filler (Future Expansion)

NOTE: New functions desired by this program have to be closely coordinated with the rental company application (ECARS or Claims Connection) requesting the new function desired in this program. All objects involved with the new function must be moved into production at the same time.

IF this program is executing on a development platform, terminate the program when returning to the calling program for deallocation of resources among many testers within an ARMS testing environment.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Files: (CRUD)

AMXBCO (-R--) ARMS Branch Claims Office Cross-reference File, using access path AMXBCOL1 by Enterprise Customer ID.

AMXREF (-R--) ARMS Cross-reference File, accessed by Vendor Transaction ID

accessed by Rental Location ID, Ticket ID

accessed by Rental Location ID, Reservation ID

ARMSPR1 (-R--) ARMS Profile File - Application Specific Flags  
ARMSPR5 (-R--) ARMS Profile File - ECARS Application Specific

Data ARMSPR6 (-R--) ARMS Profile File - CLAIMS Application Specific

Flags AMCLSTBL (-R--) ARMS Vehicle Class/Rate code file

@Embedded Data/Constants:

'AM2090V1' as the PROGRAM ID  
'B1' through 'B9' are the ARMS Identification Code for the type of information retrieval requested.

@Improvement Opportunities

1.) Convert from OPM RPG program to ILE RPG service program.

2.) Delete the 'AM2090V1' constant and use the Program Status Data Structure's left-adjusted PROGRAM ID value for the same purposes.

**ARMS Process Report**

3.) Replace the execution of the current ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

**Process**

Hierarchical numeric ID: 1.1.1.1.5.1.8  
Coded name: MILMATH  
Name: PGM Manipulate Ambiguous Dates (MILMATH)  
Comment: @Purpose:

To compare, add or subtract any passed input dates parameter field values. This program handles everything that the 'CCDATE' program does, with new standards, plus a module/program to handle addition/subtraction and comparison of ambiguous dates in different formats.

**@Operational Method:**

This program accomplishes the same thing as any of the RPG operation codes it replaces (ADD, SUBTRACT, etc., see FUNCTIONS) except that it makes century part of the equation. Basically, any dates calculated through MILMATH are expanded to include Century first and the fields in the program remain unchanged. Century is determined for all Standard Dates by comparing the year portion to 40. Century is determined for all Date Of Birth Dates by assuming no Birth dates can be in the future and no one is older than 100.

**FUNCTIONS:**

**ADDITION:** Of numbers representing dates to numbers representing dates, and of numbers to numbers representing dates.

**SUBTRACTION:** Same as Addition only MILMATH also will handle subtracting a number representing Date Of Birth from a number representing a standard date.

**COMPARISON:** Between numbers representing dates. Using COMP opcode.

**EXPANSION:** Of numbers representing dates with YY to YYYY.

Is currently executed with the following parameters:

**Action:** Input 1 Character - flag for the type of operation that MILMATH is to perform. (see FUNCTIONS.)

(YY is YYMM or YYMMDD

depending on Length 2,4 or 6.)

- (A) = YY + # of Years
- (B) = YY + YY
- (C) = YY COMPARE to YY
- (I) = YY Expand to YYYY
- (S) = YY SUBTRACT # of Years
- (T) = YY SUBTRACT YY
- (U) = YY SUBTRACT Date Of Birth

**Length:** Input 2,0 Numeric - This parameter describes the length of data being sent in DATE1 and DATE2 described below.

(2) = YY format- 2 digits.



<b>ARMS Process Report</b>
----------------------------

digits.

(4) = YYMM format- 4

(6) = YYMMDD format- 6 digits.

Date1: I/O 8,0 Numeric - This parm receives factor 1 of a calculation to be done and sends it back with century added.

Sent:	Returned:
YY	CCYY
YYMM	CCYYMM
YYMMDD	CCYYMMDD

Date2: I/O 8,0 Numeric - This parm receives factor 2 of a calculation to be done and sends it back with century added.

Sent:	Returned:
YY	CCYY
YYMM	CCYYMM
YYMMDD	CCYYMMDD

Result: Output 8,0 Numeric - This parm returns the result of the calculation if applicable.

HI Indic: Output 1 Character - This parm returns the HI indicator result of all applicable calculations.

LO Indic: Output 1 Character - This parm returns the LO indicator result of all applicable calculations.

EQ Indic: Output 1 Character - This parm returns the EQ indicator result of all applicable calculations.

**@Embedded Data/Constants:**

**Possible valid ACTION CODE values:**

'A' = Add Number of Years  
 'B' = Add Years  
 'C' = Compare Years  
 'I' = IF Compared Years  
 'S' = Subtract Number of Years  
 'T' = Subtract Year from Year  
 'U' = Subtract Date Of Birth Year from Current Year to compute age

**@Improvement Opportunities:**

Convert the programs that execute this program to ILE RPG and do all date manipulation by converting the values to the same format Date data types and performing like data type operations on the dates.

**Process**

Hierarchical numeric ID: 1.1.1.1.5.1.9  
 Coded name: CCRAZX  
 Name: PGM Retrieve GPBR Tax Detail / Cvt Sys Time to Br Time (CCRAZX)  
 Comment: @Purpose:

To retrieve a rental branch tax accounting information and convert the system time and date (currently United States Central Time Zone for all computer

ARMS Process Report
---------------------

platforms) to the rental branch's local time and date by using the TIME ZONE ADJUSTMENT field value (that is +n or -n hours from the Central Time Zone time).

@Operational Method:

This program is only executed when called by another program using two parameter fields: Group/Branch Tax Data for 215 characters, and Time Zone Conversion Data for 41 characters.

The Time Zone Conversion Data 41 characters are further redefined as:

1 alpha	Input Action Code	(possible values: A B G R T)
4 alpha	Input Group/Branch ID	
6 alpha	Input Time (hhmmss - where "ss" is optional)	
8 alpha	Input Date (mmddccyy)	
1 alpha	Input Time Format Flag	(possible values: M - Military/23-hour or S - Standard/12-hour with AM/PM)
6 alpha	Branch Military Time (hhmmss)	
6 alpha	Branch Standard Time (hhmmAM)	
6 alpha	Branch Modified Date (mmddyy)	
2 alpha	Century (cc)	
1 alpha	Program Return/Error Code	(possible values: B G R T blank)

Input Action Code dictates what this program is to return to the calling program. The codes are listed as follows:

A = Sent Group ID and Branch ID. Retrieve associated Group/Branch Tax file information and convert/adjust passed time/date according to time flag.

B = Sent Group ID and Branch ID. Only check if that Group/Branch ID value exists in Group/Branch Tax file. If not, load 'B' to the PROGRAM RETURN/ERROR CODE parameter subfield.

G = Sent Group ID only. Check if that Group ID value exists in Group/Branch Tax file. If not, load 'G' to the PROGRAM RETURN/ERROR CODE parameter subfield.

R = Sent Group/Branch ID. Retrieve associated Group/Branch Tax file record's information. Do not convert/adjust time/date. If not, load 'R' to the PROGRAM RETURN/ERROR CODE parameter subfield.

T = Time conversion only. (Not used by any program currently calling CCRAZX.)

Based on the outcome of the input codes and the data provided to the program, one of the following return codes will be generated:

(blank)	= No error
B	= Group/Branch ID record not found in the ECARS Group/Branch Tax File.
G	= Group ID not found.
R	= Group/Branch ID record not found for Group/Branch Tax file record information retrieval request.
T	= Standard Input time invalid or cannot be interpreted because none of the characters "AaPp" (for AM/PM or am/pm) are in the passed TIME field's position 5. Impossible since this program does not edit for time errors. It assumes time sent for conversion is correct.

### ARMS Process Report

Retrieves the associated GPBRTZ Group/Branch Time Zone Adjustment file record that indicates the time zone difference (in hours and fractions of hours) from United States Central time.

Program CCRAZX will expect the following implied questions from the calling programs:

"What is the branch time/date equivalent to the system time/date?"

"What are the individual government charge computation requirements for this branch?"

No editing occurs in this program - otherwise calling program is responsible for validity of input data.

A record for the branch code must exist in the BC026P file. The value of the time zone adjustment field TZADJ must be -5 to less than 19.

@Files: (CRUD) (---- signifies the file usage is only as an external data structure.)

- BC026P00 (-R--)
- GPBRTZ (-R--)
- DTPARM (----)

@Constant / Embedded Data:

AaPp - CHK array to check for a valid AM(am)/PM(pm) indication value.

@Improvement Opportunity: If the internal data structures were defined externally, then any changes to the structure would be prone to less errors and only a recompile of all programs that utilized them would need to be performed to implement the change. (NOTE: The data structure to copy into each program is in ELLIB/QDOCSRC/CCRAZX. However, some comments are out of date because they are no longer valid. This includes the specifications in the ARMS NEPs that check if the PROGRAM RETURN/ERROR CODE = "A", but no code in the CCRAZX program will load this value to that parameter subfield.)

There are several problems with the "time zone adjustment" field. BC026P:TZADJ. While these pose no problem under the current range of time zones covered by existing branches, there are basic conceptual errors which show up when the data is applied on a global basis. The first problem is the existing range, -12 to +12, specified for the field. This is a set of 25 values to be applied to 24 time zones; taking St Louis/Central time adjustment as 0, the -12 and +12 values overlap on the (global) 18th zone. Because the above set of values does not take into account the International Date Line, its use returns date errors. For systems which use Central Standard (global time zone 6) as the system time, the correct value is the offset from St Louis time; -5 to less than 19. However, the UK machine uses Greenwich as its system time (global time zone 12). In this case values from -12 to less than +12 are correct. The meaning of the field and its values needs to be given a single definition. This has been accommodated by the two fields in the newer GPBRTZ file.

GPBRTZ File Description: This file holds the time zone information for each Enterprise Rental group/branch. Time Zone information includes:

- a.) Time zone offset from the International Date Line (was Greenwich Mean Time - GMT or UTC - Universal Coordinated Time) used to determine in which time zone a branch exists. (e.g., Eastern, Central, Hawaii).
- b.) Time zone adjustment with respect to the Central Time used by the Two O'Clock programs and to print time on the tickets.
- c.) Day Light Savings Observe indicator to indicate if branch observes Daylight Savings time period (from first Sunday in April through last Sunday in October.)

<b>ARMS Process Report</b>
----------------------------

**@Notes:**

Programs EC00ATV1, EC00CNV1, EC00EXV1, ECMSGV1, AM2051V1 and module CBZ004A use this program to adjust the St. Louis (System) time value to the local group/branch's time for the transaction data set being processed. EC00ATV1 also uses this program during a different execution to retrieve the Sales Tax/Surcharge Account ID value to use for the decision of which formatted text should be used in the Branch Reservation file record's MAXIMUM AMOUNT text field.

**Process**

Hierarchical numeric ID: 1.1.1.1.5.1.10  
 Coded name: ECMSGV1  
 Name: PGM Update Electronic Message (ECMSGV1)  
 Comment: @Purpose:

To update ECARS Electronic Message database files with the received messages from the trading partner.

**@Operational Method:**

This program is called from ECARS NEP's that run in the ARMS system that process COMD01 or CBKD01 formats. It uses the transaction's passed key value received from the DQAM61V1 input data queue to access AMAPP file data and load the APPD01, ADJD01, COMD01, CBKD01 and SURD01 proprietary EDI record formats' data into the appropriate external data structures used within this program. This information is then used to update an existing or write a new ECARS Message Control File (RACMSC) record and write corresponding new Message Detail (RACMSD) file records. No data is returned to the calling program.

**@Notes:**

This program is executed with the following 3 input parameters:

6 character	Reservation ID
96 character	AMAPP Application Interface Input Transaction File key
10 character	Calling Program ID

This program does not Set On \*INLR so that it can be reexecuted without reperforming the program retrieval, memory loading/addressing and program initialization.

**@Files: (CRUD)**

AMAPP transactions file	(-R--)	Distributed APPLICATION INTERFACE input
RACBRMST	(-R--)	Branch Reservations Master File
RACMAST	(-R--)	ECARS Open Rental Contracts/Tickets Master File
RACCBIOC	(-R--)	ECARS Rental Callbacks Consolidation Control File
RACMSC	(CRU-)	Electronic Message Control File
RACMSD	(C---)	Electronic Message Detail File

**@Embedded Data/Constants:****Actual Constants:**

' - Sent by'  
 'Sent by'  
 'Surcharge Authorized: '  
 '/Day'

<b>ARMS Process Report</b>
----------------------------

```
'/Rental
' Percent
```

**Embedded Data:**

```
'EC00CMV1'
'EC00ATV1'
'AT' (Authorization Maintenance proprietary EDI transaction group type
code)
'T' (Authorization Transfer proprietary EDI transaction set type code)
'APPD01' (Internal ARMS routing header proprietary EDI record format)
'ADJD01' (Adjuster Detail proprietary EDI record format)
'CBKD01' (Callback Detail proprietary EDI record format)
'COMD01' (Comment proprietary EDI record format)
'SURD01' (Surcharge Approved for Direct Billing proprietary EDI record
format)
'R' (Rental Document ID Reservation Type Prefix)
'D' (Rental Document ID Rental Contract/Ticket Type Prefix)
'*' (LastName*FirstName* delimiter)
'ARMS' (Record Created By Employee)
'D' (Per Rental-Daily Amount Surcharge)
'L' (Per Rental-Lump Amount Surcharge)
'P' (Per Rental-Percentage Surcharge)
'+' (Surcharge Amount Positive Sign)
'-' (Surcharge Amount Negative Sign)
'05' (Government Surcharge Type Code)
'I'
'E'
'Y' (Yes)
'N' (No)
'T'
'A'
'M' ('CCRAXZ' program execution input parameter for Action Code)
```

**@Improvement Opportunities:**

Convert this OPM RPG program to an ILE RPG service program comprised of ILE RPG modules.

**Process**

```
Hierarchical numeric ID: 1.1.1.1.5.1.11
Coded name: CCRS10
Name: PGM Print Reservation Detail (CCRS10)
Comment: @Purpose:
```

To notify and alert the rental branch that a rental authorization has been received from the rental management trading partner. Furthermore, display specific information about the reservation: Renter's name; address; phone number; Insurance replacement information and any special requests (Vehicle-Van, requires pickup date/time).

**@Operational Method**

- Retrieve branch reservation detail record
- IF reservation was created thru ARMS
  - Print fields and headings. Calculate renter pay percent
- Convert flag fields to constants (pickup status, loss type)
- Build surcharge print lines

ARMS Process Report
---------------------

- Retrieve last 10 messages ONLY  
 - Override the printer file to the Group/Branch output queue. Change  
 formtype to PLAINPAPER

@Files: (CRUD)

RACBRMST	(-R--)
RACPAYOP	(-R--)
LIFEM#1	(-R--)
ECSURCHG	(-R--)
RACMSG1	(-R--)

### Process

Hierarchical numeric ID: 1.1.1.1.5.2

Coded name:

Name: AUT Interface with Claims Connection (AT/CN/EX)

Comment: @Definition: The automatic process by which the Claims Connection database files are updated as a result of a request sent by the Rental Management Trading Partner (AT, CN, EX,).

### Process

Hierarchical numeric ID: 1.1.1.1.5.2.1

Coded name: CC00ATV1

Name: PGM Interface CC with Rental Authorization (CC00ATV1)

Comment: @Purpose:

To update the claims connection database with any authorization information received from the trading partner.

@Operational Method:

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of 'ATC'.

- IF the receive data queue entry is a shutdown request, send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN

- Open programs files, if not previously opened.

- Read the transaction data set records from AMAPP and populate the associated record format.

- IF authorization is a change request and there's no ticket update the reservation.

- IF authorization is a change request and there is a ticket, update the ticket. Communicate this change to ARMS by forwarding a data queue entry to DQAM60V1.

- IF authorization is an add or transfer request and there's no ticket, create the reservation. Communicate this change to ARMS by forwarding a data queue entry to DQAM60V1.

- IF authorization is an add or transfer request and there is a ticket,  
 create the ticket. \*\*\*Can't happen.

ENDIF

@Improvement Opportunity:

ARMS Process Report
---------------------

- Use program status data structure to retrieve program name.

@Files (CRUD)

ZLIFM#1	(-R--)
CCEMP	(-R--)
CUSTMAST	(-R--)
DROFLF1	(-R--)
CCEXTCTL	(C---)
CCEXTDTL	(C---)
AMAPP	(C---)
CCINST	(C---)
CCSURDTL	(CR-D)
CCRES	(CRU-)
CCNUM	(-RU-)
CCMASTER	(CRU-)

@Notes:

- This programs only runs on the Central machine.
- The request(s) sent by the Rental Management Trading Partner can either be an authorization adds(AT-Add) or authorization changes (AT-Chg). There are three types of authorization adds: 1) An unsolicited authorization. 2) Authorization on a prior vendor authorization request. 3) Authorization for an open ticket (Vendor over-ride)
- Presently there is a business limitation where you can not authorize an existing open ticket.

#### Process

Hierarchical numeric ID: 1.1.1.1.5.2.2

Coded name: CC00EXV1

.Name: PGM Interface CC with Rental Extension/Termination of Authorization (CC00EXV1)

Comment: @Purpose:

To update the claims connection database with any rental extension information received from the trading partner in the function group type 'EX'.

#### @Operational Method

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of 'EXC'.
- Open programs files, if not previously opened.
- IF the receive data queue entry is a shutdown request, send that request to data queue DQANDST to inform the distributed send program it has ended.
- IF the receive data queue entry is a non-shutdown request THEN
  - Read the transaction data set records from AMAPP and populate the associated record format.
  - Retrieve ARMS trading partner profile information through the use of ARMS data retrieval program(AM2090V1)
  - IF extension is an extension request THEN
    - Calculate new extension date.
    - Generate associated text message for the history file.
    - Re-calculate bill-to and bill-from dates.
  - ENDIF
  - IF extension is a termination request THEN

ARMS Process Report
---------------------

- Validate termination date received
- Generate associated text message for the history file.

ENDIF

ENDIF

@Files:

```

CCEXTCTL      (CRU-)
CCEXTDTL      (C---)
CCMASTER      (-RU-)
AMAPP          (-R--)
REQSTCTL      (CRU-)

```

@Improvement Opportunity

- Use program status data structure to retrieve program name.

@Constants

```

CONST('Rental Extended Until:')
CONST('Rental Terminated On:')
CONST('OVRDBF FILE(CCMaster) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCMaster)')
CONST('OVRDBF FILE(CCEXTCTL) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCEXTCTL)')
CONST('OVRDBF FILE(CCEXTDTL) WAITRCD(*IMMED)')
CONST('DLTOVR FILE(CCEXTDTL)')

```

@Notes:

- This program only runs on the centralized finicial system(Central).
- The request(s) sent by the Rental Management Trading Partner can either be an extension or termination.

## Process

Hierarchical numeric ID: 1.1.1.1.5.2.3

Coded name: CC00CNV1

Name: PGM Interface CC with Rental Cancellation of Authorization (CC00CNV1)

Comment: @Purpose:

To process cancelation requests received from the trading partner in the functional group type 'CN'.

@Operational Method

- Wait indefinitely for the next DQAM61V1 data queue entry with a key of 'CNC'.

- Open programs files, if not previously opened.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Read the transaction data set records from AMAPP and populate the associated record format.
  - IF record not available THEN



<b>ARMS Process Report</b>
----------------------------

```

- send transaction request back to DQAM61V1.
ELSE
- Delete reservation record from file(CCREs).
- Generate associated text message for the history file.
ENDIF

```

ENDIF

@Improvement Opportunity:

- Use program status data structure to retrieve program name

@Constants/Embedded Data

FILE( ) WAITRCD(\*IMMED)

@Files:

```

AMAPP      (-R--)
CCINST     (CR--)
CCRES      (-R-D)
CCDIRBIL   (-R-D)

```

#### Process

Hierarchical numeric ID: 1.1.1.1.5.2.43

Coded name: VFYOBJ

Name: PGM Verify Objects Existence (VFYOBJ)

Comment: @Purpose:

To determine if the specified object exists on the host platform.

@Operational Method:

- Check for the objects existence, return a code of '1' when object not found.

@Notes:

- This program is used by programs written in languages other than the CL to take advantage of the CHKOBJ CL command.

#### Process

Hierarchical numeric ID: 1.1.1.1.6

Coded name:

Name: AUT Generate Vendor Authorization Maintenance Request (AC/RA/RC/RE/RN/TR/VM)

Comment: @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats(data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner. (This includes ECARS and Claims Connection)

#### Process

Hierarchical numeric ID: 1.1.1.1.6.1

Coded name:

Name: AUT Generate Claims Connection Authorization Maintenance Request (AC/RA/RC/RE/RN/TR)

### ARMS Process Report

**Comment:** @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats(data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner.

#### Process

Hierarchical numeric ID: 1.1.1.1.6.1.1  
 Coded name: CC00TRV1  
 Name: PGM Generate CC Transfer Request (CC00TRV1)  
 Comment: @Purpose:

To generate authorized reservation transfer (functional group type AT-Transfer) group type transaction data sets for sending to another Rental application system in response to a Claims Connection Rental System user requesting a transfer of an authorized reservation to an Enterprise Rent-A-Car rental group's rental functional site/branch.

#### @Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'TRC' (for Claims Connection) in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request, then from the data queue entry's Rental Location's Group ID and the Reservation ID passed, retrieve record from the CCRES rental reservation file.
- Populate the associated record formats' fields as follows:
  - Load the retrieved reservation's Transferred To Location Group/Branch ID to the COMD01 Comment Format's Comment Text with the COMTX data structure preformatted comment text immediately after the word "to".
  - Populate the ADJD01 data structure with the reservation's Bill-To Attention Name, Phone and Extension, the Bill-To Street Address, City, State and Postal Code.
  - If adjustor name or phone/extension fields of the ADJD01 record format are blank or zeroes, then retrieve the reservation's associated instructions file record's for sequence 001 and extract and load this adjustor information from that instruction text line.
  - Load every non-blank reservation's associated CCINST Instruction file record's Instruction text field value to a COMD01 comment record format's comment field.
  - Load the RNTD01 Renter Detail and AUTD01 Authorization Detail record formats' fields with the associated fields' values of the retrieved CCRES file record.
  - If the retrieved CCRES file record's Repair Shop Name is not blank, then load that field value and its associated contact name and telephone number to the RPRD01 Repair Detail record format.
  - Load the retrieved CCRES file record's Renter's Car (Vehicle being replaced) Year (of Manufacture) and associated Make/Model and Date of Loss field values to their associated fields in the RNTD02 Renter Detail #2 record format.
  - Load the retrieved CCRES file record's Work Phone, Home Phone and Social Security Number field values to their associated fields in the RNTD02 Renter Detail #2 record format's fields.

### ARMS Process Report

- If this transfer is NOT going to Claims Connection, transfer each associated electronic message detail file records field value to their associated CBKD01 record format's fields.

- If attempt to retrieve the associated CCEXTCTL Callback Control file record is successful and its record type is "R" (Reservation), transfer each of its Callback Type Codes and their associated Status Code field values to their associated CBKC01 Callback Control record format's fields. Likewise, if such a callback control format was loaded, then for every associated CCEXTDTL Callback Detail file record retrieved, transfer its Note #1 and Note #2 field values, along with the note's type original creation date, time, Employee ID and Program ID, to their associated CBKD01 Callback Detail record format's fields.

- Delete all associated CCSURDTL Surcharge Detail file records.
- Delete all associated CCEXTDTL Callback Detail file records.
- Delete the associated CCEXTCTL Callback Control file record.

- Write all of the non-blank record formats of the authorized reservation transfer (AT-Transfer) group type transaction data set to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Sends the written authorized reservation transfer (AT-Transfer) group type transaction data set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files: (CRUD) [When is shown as (----), this is used as an externally-defined data structure.]

```

- ARMSDATA      (----)
- ARMSKEY       (----)
- DQAM60V1      (----)
- APPD01V1      (----)
- ADJD01V1      (----)
- AUTD01V1      (----)
- COMD01V1      (----)
- INSD01V1      (----)
- RNTD01V1      (----)
- RNTD02V1      (----)
- RFRD01V1      (----)
- CBKC01V1      (----)
- CBKD01V1      (----)
- $#EMACH       (----)
- CCINST        (-R--)
- CCRES         (-R--)
- CCSURDTL      (-R-D)
- CCEXTCTL      (-R-D)
- CCEXTDTL      (-R-D)
- ANDIST        (C---)

```

@Embedded Constant:

COMTX 'ARMS reservation transferred from Claims Connection to

@Notes:

**ARMS Process Report**

- This program does not use the INSD01 format, but the EC00TRV1 program does.
- This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the distributed host computer system platform where Claims Connection Rental System application database resides. This program is currently submitted by the ARMS Start-Up Job (CLL810) when executed on the system known as "CENTRAL" host computer system platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input.

**Process**

Hierarchical numeric ID: 1.1.1.1.6.1.2

Coded name: CC00RNV1

Name: PGM Generate CC Rental Notification (CC00RNV1)

Comment: @Purpose: To generate an opening or closing rental notification, functional group type 'RN' transaction data sets for sending to the Trading Partner host application system in response to a renter starting or ending the rental contract with a third-party rental vendor other than Enterprise Rent-A-Car.

@Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RNC' in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request with an action code of '3', then retrieve the record from the CCMaster open rental contract/ticket file and populate the associated record formats' fields, leaving the End Date and End Time fields initialized to zeroes.
- If the received data queue entry is a non-shutdown request with an action code of '4', then retrieve the record from the CCCLSC closed rental contract/ticket file and populate the associated record formats' fields, leaving the Start Date and Start Time fields initialized to zeroes.
- Load the OFFD01V1 format with the passed data queue entry's contract/ticket ID and its location, along with the retrieved shipping name, street address, city, state, postal code and toll-free telephone number for the Claims Connection office.
- Write Rental Notification (RN) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.
- Sends the written Rental Notification group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.
- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files:

<b>ARMS Process Report</b>
----------------------------

```

- OFFDRB      (-R--)
- CCMaster    (-R--)
- CCCLSC      (-R--)
- ANDIST      (CRU-)

```

**@Embedded Constant:**

'7680' is the sole Group/Branch ID key value used to access its street/city/state address from the Branch Office Directory file (OFFDRB) at the beginning of the program.

8002277800 as the overriding telephone number loaded into all ELCD01:ELCPHN fields.

**Process**

Hierarchical numeric ID: 1.1.1.1.6.1.3

Coded name: CC00RCV1

Name: PGM Generate CC Request for Authorization Cancellation (CC00RCV1)

Comment: @Purpose:

To generate a requests for authorization cancellation, functional group type 'RC', transaction data sets for sending to the Trading Partner host application system.

**@Operational Method:**

- Receive the next DQAM60V1 data queue entry with the key value of 'RCC' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request with an action code of '6', then load the APPD01V1 record format's Add/Change/Transfer Code with 'P', else load it with a blank. (Unsure this is currently needed since AM0046V1 does not reference any actions base on APPCDE with a 'P' or blank value.)

- Write request for authorization cancellation (RC) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Sends the written request for authorization cancellation (RC) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

**@Notes:** This program is only executed as an ARMS never-ending batch program in the ARMS subsystem only on the host computer system platform where Claims Connection Rental System application database resides. This program is submitted by the ARMS Start-Up Job (CLL810) when executed on the "CENTRAL" host computer platform. This never-ending batch program ends normally when a shutdown data queue entry is received as input.

**@Files:**

ARMS Process Report
---------------------

- ANDIST (C---)

@Improvement Opportunity:

There is no real difference in this program and the CC00RCV1 program. With a minor change in the retrieved data queue entry key value's third position ("B" vs. "C") and the loading of the APPSRC field with that same value, there is no differences between these programs.

Process

Hierarchical numeric ID: 1.1.1.1.6.1.4

Coded name: CC00REV1

Name: PGM Generate CC Request for Authorization Extension (CC00REV1)

Comment: @Purpose:

To generate rental extension request, functional group type, 'RE' transaction data set for sending to the Trading Partner.

@Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'REC'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Retrieve the necessary data base records
  - IF extension request criteria met THEN
    - Build the record formats, write the record to the distribution transaction file (ANDIST).
    - Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

ENDIF

- Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

@Files:

CCEXTCTL	(-R--)
CCEXTDTL	(-R--)
CCMASTER	(-R--)
REQSTCTL	(CRU-)
ANDIST	(C---)

@Notes:

- This program only runs on the centralized finicial system(Central).
- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - VEDD01 - Vendor Extension format
  - COMD01 - Comment detail format
  - RPRD01 - Repair detail format

<b>ARMS Process Report</b>
----------------------------

**Process**

Hierarchical numeric ID: 1.1.1.1.6.1.5

Coded name: CC00RAV1

Name: PGM Generate CC Request for Authorization Maintenance (CC00RAV1)

Comment: @Purpose:

To generate an Authorization Request, functional group type, 'RA' transaction set for sending to the Trading Partner.

**@Operational Method**

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'RAC'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN

- Using the action code set the routing for the transaction

- IF request is on a ticket THEN

- IF ticket is 'open' then

- Retrieve the open ticket data base records (CCMASTER)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction

file(ANDIST).

(Only build RATD01 record format if trading

partner

profile indicates it).

ELSE

- Retrieve the closed ticket data base records (CCCLSC)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction

file(ANDIST).

(Only build RATD01 record format if trading

partner

profile indicates it).

ENDIF

ELSE

- Retrieve the reservation data base records (CCRES)

- Retrieve the Rental Management Trading Partner profile specifics

- Build the applicable record format(s), writing the records to the distribution transaction file(ANDIST).

ENDIF

- Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

- Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

@Files:

ARMS Process Report
---------------------

CCMASTER	(-R--)
CCCLSC	(-R--)
CCRES	(-R--)
CCINST	(-R--)
OFFDRB	(-R--)
ANDIST	(C--)

## @Notes:

- This program only runs on the centralized financial system(Central).
- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - AUTD01 - Authorization detail format
  - CMD01 - Comment detail format
  - RATD01 - Rate detail format
  - RNTD01 - Renter detail 1 format
  - RNTD02 - Renter detail 2 format
  - RPRD01 - Repair detail format
  - INSD01 - Insured detail format
  - ELCD01 - Arms vendor location

## Process

Hierarchical numeric ID: 1.1.1.1.6.1.11  
 Coded name: CC00ACV1  
 Name: PGM Generate CC Authorization Confirmation (CC00ACV1)  
 Comment: @Purpose:

To generate an Authorization Confirmation, functional group type (AC), transaction set that informs the Trading Partner of the authorization confirmation number and the Enterprise location that is handling the insurance replacement.

## @Operational Method:

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'ACC'.
- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.
- IF the receive data queue entry is a non-shutdown request THEN
  - Construct the necessary record format, write the record to the distribution transaction file(ANDIST)
  - Send a data queue entry on to the Distributed Send to Centralized data queue(DQANDST).
  - Send a data queue entry to Dispatch Data Queue(DQAM60V1) with a key of 'DIS' to finish the request.

## @Improvement Opportunity:

- Use program status data structure to retrieve program name

## @Embedded Constant:

'7680' - default location  
 '8002277800' - default location phone#  
 'CC00ACV1' - Program name



**ARMS Process Report****Files:**

OFFDRB (-R--)  
ANDIST (CRU-)

**@Notes:**

- This program only runs on the CENTRAL machine.
- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - ELCD01 - Arms vendor location

**Process**

Hierarchical numeric ID: 1.1.1.1.6.2

Coded name:

Name: AUT Generate ECARS Authorization Maintenance Request (AC/RA/RC/RE/RN/TR/VM)

Comment: @Definition: The automatic process of dispatching data queue entries for appropriate programs to generate the required record formats(data set) to be sent to a Rental Management Trading Partner in response to an action by the Rental System or in response to some communication received from the Rental Management Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.1.1.6.2.3

Coded name: EC00VMV1

Name: PGM Generate EC Vendor Message (EC00VMV1)

Comment: @Purpose:

To generate a vendor message functional group type, 'VM', transaction data sets for sending an electronic message from the rental agent to the trading partner.

**@Operational Method:**

- Receive the next DQAM60V1 data queue entry with the key value of 'VMB' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request, then load the APPD01V1 record format and for every RACMSD record read by the passed Group/Branch ID and the non-blank Ticket ID prefixed with a 'D' (or if ticket is blank the passed Reservation ID prefixed with a 'R'), load a COMD01V1 record format and update the RACMSD file record's MESSAGE STATUS field with the value 'O' (Output).

- Write vendor message (VM) group type transaction data set records to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written vendor message (VM) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

### ARMS Process Report

- Send the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

@Files: (CRUD) (---- signifies usage as an external data structure)

- RACMSD1 (-RU-)
- ANDIST (C---)
- APPD01V1 (----)
- COMD01V1 (----)
- DQAM60V1 (----)

@Embedded Constant: Program name 'EC00VMV1'. (Could be replaced by use of the program status data structure's subfield for program.)

#### Process

Hierarchical numeric ID: 1.1.1.1.6.2.4

Coded name: EC00RCV1

Name: PGM Generate EC Request for Authorization Cancellation (EC00RCV1)

Comment: @Purpose:

To generate a request for authorization cancellation functional group type, 'RC', transaction data sets for sending to the trading partner.

#### @Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RCB' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request with an action code of '6', then load the APPD01V1 record format's Add/Change/Transfer Code with "P"; else load it with a blank. (Unsure this is currently needed since AM0046V1 does not reference any actions base on APPCODE with a "P" or blank value.)

- Write request for authorization cancellation (RC) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written request for authorization cancellation (RC) group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Send the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatched task has been completed.

@Files: (CRUD)

- ANDIST (C---)

### ARMS Process Report

@Improvement Opportunity: There is no real difference between this program and the program, CC00RCV1. With a minor change in the retrieved data queue entry key value's third position ("B" vs. "C") and the loading of the APPSRC field with that same value, there is no differences between these programs.

#### Process

Hierarchical numeric ID: 1.1.1.1.6.2.5  
 Coded name: EC00ACV1  
 Name: PGM Generate EC Authorization Confirmation (EC00ACV1)  
 Comment: @Purpose:

To generate an Authorization Confirmation, functional group type, 'AC', transaction data set that informs the Rental Management Trading Partner of the authorization confirmation number and the Enterprise location that is handling the insurance replacement.

#### @Operational Method

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'ACB'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Construct the necessary record format, write the record to the distribution transaction file(ANDIST)
  - Send a data queue entry on to the Distributed Send to Centralized data queue(DQANDST).
  - Send a data queue entry to Dispatch Data Queue(DQAM60V1) with a key of 'DIS' to finish the request.

@Files (CRUD)  
 OFFDRB (-R--)  
 ANDIST (C---)

#### @Notes

- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - ELCD01 - Arms vendor location

#### @Improvement Opportunity

- Use program status data structure to retrieve program name

@Constant/Embedded Data  
 'EC00ACV1' - Program name

#### Process

Hierarchical numeric ID: 1.1.1.1.6.2.6  
 Coded name: EC00REV1  
 Name: PGM Generate EC Authorization Extension Request (EC00REV1)  
 Comment: @Purpose:

To generate an Extension Request, functional group type 'RE', transaction data set for sending to the Rental Management Trading Partner in response to the

### ARMS Process Report

Rental System speaking with the bodyshop and changing the contract's estimated completion date.

#### @Operational Method

'REB'.  
- Wait indefinitely for the next DQAM60V1 data queue entry with a key of

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Retrieve the necessary data base records
  - IF extension request criteria met THEN
    - Build the record formats, update the database records, write the record to the distribution transaction file (ANDIST).
    - Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

ENDIF

- Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

#### @Notes

- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - VEDD01 - Vendor Extension format
  - COMD01 - Comment detail format
  - RPRD01 - Repair detail format

#### @Files (CRUD)

CB007P00	(-R--)
CB012P01	(-RU-)
RACMSG2	(-RU-)
RACMAST	(-R--)
RACSMAS	(-R--)
ECEXTCTL	(CRU-)
ANDIST	(C---)

#### @Improvement Opportunity

- Use program status data structure to obtain the program name.

#### @Constants/Embedded Data

- 'EC00REV1' - pgm name

#### Process

Hierarchical numeric ID: 1.1.1.1.6.2.7

Coded name: EC00RAV1

Name: PGM Generate EC Authorization Maintenance Request (EC00RAV1)

Comment: @Purpose:

To generate an Authorization Request functional group type, 'RA' transaction data set for sending to the Rental Management Trading Partner in response to the Rental System creating/changing a rental contract.

## ARMS Process Report

## @Operational Method

```

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'RAB'.

- IF the receive data queue entry is a shutdown request, close program files and send
that request to data queue DQANDST to inform the distributed send program it has
ended.

- IF the receive data queue entry is a non-shutdown request THEN
  - Attempt to retrieve the reservation data base records (RACBRMST)
    - IF request is on a ticket THEN
      - Retrieve the open and closed ticket database records
        (RACMAST, RACSMAS, RACCLSC, RACSCLC)
      - Using the action code set the routing for the transaction
    - Retrieve the Rental Management Trading Partner profile specifics
      - Build the applicable record format(s) from the database files, writes the
records to the distribution transaction file. (Only build RATD01 record format if
profiled.)
      - Send a data queue entry on to the Distributed Send to
Centralized data queue (DQANDST).
      - Send a data queue entry to Dispatch Data Queue
(DQAM60V1) with key of 'DIS' to finish the request.
    ENDIF
  - IF request is on a reservation THEN
    - Using the action code set the routing for the
      - Retrieve the Rental Management Trading Partner
profile specifics
      - Build the applicable record format(s), writing the
records to the distribution transaction file.
(Only build RATD01 record format if profiled)
    - Send a data queue entry on to the Distributed Send to
Centralized data queue (DQANDST).
      - Send a data queue entry to Dispatch Data Queue
(DQAM60V1) with a key of 'DIS' to finish the request.
    ENDIF
  ENDIF
transaction
- Send a data queue entry on to the Distributed Send to
Centralized data queue (DQANDST).
- Send a data queue entry to Dispatch Data Queue
(DQAM60V1) with a key of 'DIS' to finish the request.
ENDIF
ENDIF

```

```

@Files          (CRUD)
RACMAST          (-R--)
RACSMAS          (-R--)
RACBRMST         (-R--)
RACBRIST         (-R--)
RACINS           (-R--)
RACBRWRT         (-R--)
OFFDRB          (-R--)
RACCLSC          (-R--)
RACSCLC          (-R--)
ANDIST           (C---)

```

## Process

```

Hierarchical numeric ID: 1.1.1.1.6.2.8
Coded name: RTVEMCHA
Name: PGM Retrieve Machine Attributes (RTVEMCHA)
Comment: @Purpose:

```

To retrieve the machine attributes for the machine that this program is running on.

# ARMS Process Report

## @Operational Method:

- Retrieve the current system name (factor in emulation).
- Retrieve attributes associated to host platform.
- Return data elements to calling program.

## @Notes:

- \$EMCAH is an external data structure used as a template for the data passed to and from.

@Files (CRUD)  
MACHID#1 (-R--)  
GACTL (-R--)

## Process

Hierarchical numeric ID: 1.1.1.1.6.2.17

Coded name: AM1070V1

Name: PGM Retrieve ARMS Vehicle Class and Rate Designation (AM1070V1)

Comment: @Purpose:

To retrieve the associate rental rate amount for a given vehicle class code, OR, to retrieve an associated vehicle class code for a specified rental rate amount.

## @Operational Method:

The input parameters are: Group ID, Branch ID, Vehicle Class Code (Optional), Daily Rental Rate Amount (Optional), Effective Date (Optional - Else current system date will be used), and Control Mode ('T' = for rental contract Ticket; 'R' = for rental Reservation).

- IF an Effective Date value is not received, then retrieve and use system date for Effective Date.

- IF request is on a rental contract/ticket (Control Mode = 'T'), a valid Vehicle Class Code will be returned for the effective date sent. (Daily Rental Rate Charged Amount will always have a value when sent since it is an open or closed rental contract.)

-- Validate the rate received by comparing effective date to the rate date ranges. Then retrieve the associated vehicle class code for that rate in the National Reservations Standard Rates (NRXRAT1) file, for 'I' Insurance type customers only by Group/Branch IDs, Daily Rental Rate Amount and Effective Start Date, comparing the Effective Date to being on or within the rental rate amount's Effective Start and Stop Date ranges.

-- IF all of the records are read within a Group/Branch ID and no valid records are found, then retrieve a vehicle class code by reading backwards through NRXRAT1 until a valid record is found.

--- IF no valid record is found, then return blanks in the Vehicle Class Code.

--- IF a valid record is found, then verify its vehicle class code is valid by retrieving an associated record in National Reservations Car Type (NRCTY) file.

--- A valid record in the NRCTY file is a record where the Effective Date is within the Effective Start and Stop dates, inclusive.

--- IF rate and vehicle class do not match, then retrieve the class for the previous rate with a valid effective date.

- ENDIF

- IF request is on a reservation (Control Mode = 'R')

### ARMS Process Report

-- When Vehicle Class Code and Daily Rental Rate Amount (Quoted/Authorized) has been received

--- Only validate that the Daily Rental Rate Amount (Quoted) received is compatible with the Vehicle Class Code by retrieving the daily rental rate amount's associated vehicle class code for that rate in the National Reservations Standard Rates (NRXRAT1) file, for "I" Insurance type customers only by Group/Branch IDs, Daily Rental Rate Amount and Effective Start Date, comparing the Effective Date to being on or within the rental rate amount's Effective Start and Stop Date ranges. Then find the associated vehicle class code for that daily rental rate amount.

--- IF the Daily Rental Rate Amount's Vehicle Class Code and the passed Vehicle Class Code do not match, then do not return either the rate nor vehicle class code, replace them with blanks.

- When Vehicle Class Code received, the associated Rate Amount will be returned if a valid record is found within the effective date.

- Validate the Vehicle Class Code received, then find the associated vehicle daily rental rate for that class.

IF rate and vehicle class do not match, then do not return either rate or class.

\*\* This option is not return values when found. Issue will be written and submitted.

- When Rental Rate Amount received, an attempt will be made to find a valid vehicle class.

- Validate the rate received by comparing effective date to the rate date ranges. Then find the associated vehicle class code for that rate.

IF rate and vehicle class do not match, then do not return either rate or class.

ENDIF

#### @Notes:

When indicating that Vehicle Class Code and Rental Rate Amount not returned, it means that these fields will be cleared to their lowest possible values, blanks and zeroes respectively.

Sometimes there are multiple vehicle types for any given Group ID, Branch ID, and Vehicle Class Code. However, this program has been changed to allow a rate to be returned correctly when this situation occurs.

@Files: (CRUD)

NRXRATES (-R--) by both logical access paths: NRXRAT1 and NRXRAT2.

NRCTYP (-R--) by both logical access paths: NRCTY and NRCTL2.

#### Process

Hierarchical numeric ID: 1.1.1.1.6.2.20

Coded name: ECCBBSM

Name: PGM Maintain Consolidated Body shop Callbacks Control File for ARMS (ECCBBSM)

Comment: @Purpose: To maintain the consolidated body shop callback information.

#### @Operational Method:

- This program accepts a parameter list that contains specific data elements as it relates to the callback (group, branch, ticket #, type, status, completion date, shop info, etc.) and updates the rental system (ECARS) data base file.

<b>ARMS Process Report</b>
----------------------------

passed:

- Determine tasks to be performed based on the control field value

calling program

- 'X' = Exit program
- 'C' = Retrieve callback information from database and return to
- 'D' = Delete callback
- 'K' = Check if callback exists
- 'L' = Lock callback from other users
- 'U' = Update callback with the passed information

**@Files:**

RACCBBS            (CRUD)

**Process**

Hierarchical numeric ID:        1.1.1.1.6.2.22

Coded name: UTS001A

Name: PGM Generate Century for DOB (UTS001A)

Comment:    @Purpose: To convert a six character date of birth to an eight character date with the correct century.

**@Operational Method:**

- this program accepts a parameter list from which a date is received as YYMM and converted to CCYYMMDD.
- Set base line date (current date minus 1)
- Load current date century into output date of birth century.
- IF the passed date of birth with the concatenated century is greater than the base line date, subtract 1 from the date of birth century.
- Return converted (CCYYMMDD) date of birth.

**Process**

Hierarchical numeric ID:        1.1.1.1.6.2.24

Coded name: EC00TRV1

Name: PGM Generate EC Transfer Request (EC00TRV1)

Comment:    @Purpose:

To generate authorized reservation transfer (AT-Transfer) functional group type transaction data sets for sending to another Rental application system in response to a rental user transferring an authorized reservation to a different rental functional site (group).

**@Operational Method:**

- Receive the next DQAM60V1 data queue entry with the key value of 'TRB' (Transfers from Enterprise Branch Rental System) in infinite wait mode.
- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.
- If the received data queue entry is a non-shutdown request, then from the data queue entry's Rental Location's Group ID and the Reservation ID passed, retrieve record from the RACBRMST rental reservation file.



### ARMS Process Report

- Populate the associated record formats' fields as follows:
  - IF the retrieved reservations closing Group/Branch ID is '7680' (Claims Connection), then load the COMD01 Comment Format's Comment Text with the COM002 data structure preformatted comment text and insert the reservation's Pickup Group/Branch immediately after the word "from", else load the COM003 data structure preformatted comment text with the inserted reservation's Pickup Group/Branch ID immediately after the word "from" and insert the reservation's closing Group/Branch ID after the word "to".
  - Populate the ADJD01 data structure with the reservation's Bill-To Attention Name, Phone and Extension, the Bill-To Street Address, City, State and Postal Code.
    - IF adjustor name or phone/extension fields of the ADJD01 record format are blank or zeroes, then retrieve the reservation's associated instructions file records for sequence 001 and extract and load this adjustor information from that instruction text line.
    - Load every non-blank reservation's associated instruction file record that has a sequence number 002 through 096 as a COMD01 comment record format.
    - Load the associated RACINS Renter Insurance Coverage file record's Policy ID and Policy Expiration date to the INSD01 Insured record format.
    - Load the retrieved RACBRIST file's 996 - 998 record's Insured name, address to the INSD01 record format fields.
    - IF the INSD01 record format's Policy ID and its expiration date is blank/zero, then load them from the RACBRIST file records' Policy ID and Expiration Date.
    - Load the RNTD01 Renter Detail and AUTD01 Authorization Detail record formats' fields with the associated fields' values of the retrieved RACBRMST file record.
    - If the retrieved RACBRMST file record's Repair Shop Name is not blank, then load that field value and its associated contact name and telephone number to the RPRD01 Repair Detail record format.
    - Attempt to retrieve the associated RACBRWRT Reservation Pre-Write file record and load its Date of Birth field value to the same field in the AUTD01 record format's same field.
    - If the retrieved RACBRMST file record's Renter's Car (Vehicle being replaced) Year (of Manufacture) is non-blank, then load that field value and its associated Make/Model and Date of Loss field values to their associated fields in the RNTD02 Renter Detail #2 record format.
    - If the retrieved RACBRMST file record's Work Phone, Home Phone or Social Security Number field values are not zeros or blanks, then also load those values to their associated fields in the RNTD02 Renter Detail #2 record format.
    - For every associated RACBDET Callback Detail file record retrieved, transfer its Note #1 and Note #2 field values, along with the note's type original creation date, time, Employee ID and Program ID, to their associated CBKD01 Callback Detail record format's fields.
    - If this transfer is NOT going to Claims Connection, transfer each associated electronic message detail file records field value to their associated CBKD01 record format's fields.
    - If attempt to retrieve the associated CB007P00 Callback Control file record is successful, transfer each of its Callback Type Codes and their associated Status Code field values to their associated CBKC01 Callback Control record format's fields.
  - Write all of the non-blank record formats of the authorized reservation transfer (AT-Transfer) group type transaction data set to ANDIST transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.
  - Sends the written authorized reservation transfer (AT-Transfer) group type transaction data set record formats' key value to the ANDIST output file's associated DQANDST data queue.

ARMS Process Rep rt
---------------------

- Delete all associated ECSURCHG Surcharge Detail file records.
- Delete all associated RACMSC Electronic Message Control file records.
- Delete the associated CB012P Electronic Message Detail file record.
- Delete the associated RACCBK3 Callback Consolidation Skeleton file record.
- Delete all associated RACCBDT Callback Detail file records.
- Delete all associated RACPAYOP Open Payments file records.
- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

## @Notes:

Do not write INSD01 format for a transfer of an authorized reservation for a claimant, since the AT-Add had updated the claimant's insurance information to create a RACINS (ECR Renter's Insurance Detail) record without loading the INSURED NAME and CITY/ST/ZIP. However, when the branch reservation is transferred, this program loaded the INSD01 record format with this insured information from RACINS but the INSURED NAME, CITY/ST/ZIP and RECORD FORMAT ID were not loaded since they are blanks. This is because that information is not available at the time of the transfer. This is to prevent the AT-Transfer from being rejected in AM0025V1.

@Files: (CRUD) (When is shown as (----), this is used as an externally-defined data structure.)

- ARMSDATA	(----)
- ARMSKEY	(----)
- DQAM60V1	(----)
- APPD01V1	(----)
- ADJD01V1	(----)
- AUTD01V1	(----)
- COMD01V1	(----)
- INSD01V1	(----)
- RNTD01V1	(----)
- RNTD02V1	(----)
- RPRD01V1	(----)
- CBKC01V1	(----)
- CBKD01V1	(----)
- \$EMACH	(----)
- RACBRMST	(-R--) Opened with the OVRDBF
- RACBRIST	(-R--) Opened with the OVRDBF
- RACINS	(-R--) Opened with the OVRDBF
- RACBRWRT	(-R--) Opened with the OVRDBF
- RACCBK3	(-R-D) Opened with the OVRDBF
- RACCBIOC	(-R-D) Opened with the OVRDBF
- RACPAYOP	(-R-D) Opened with the OVRDBF
- ECSURCHG	(-R-D) Opened with the OVRDBF
- CB007P00	(-R-D) Opened with the OVRDBF
- RACCBDT	(-R-D) Opened with the OVRDBF
- RACMSC	(-R-D) Opened with the OVRDBF
- CB012P00	(-R-D) Opened with the OVRDBF
- RACPAYCL	(C---) Opened with the OVRDBF
- ANDIST	(C---) Opened with the OVRDBF

@Constants as Compile-Time Data:

OVRDBF FILE( ) WAITRCD(\*IMMED)

ARMS Process Report		
---------------------	--	--

YES	C	CONST('Y')
NO	C	CONST('N')
OFF	C	CONST('0')
ON	C	CONST('1')
Process/shutdown codes		
TR	C	CONST('TR')
SD	C	CONST('SD')
TSTDWN	C	CONST('*DOWN')
Format Names		
APPDF1	C	CONST('APPD01')
ADJDF1	C	CONST('ADJD01')
AUTDF1	C	CONST('AUTD01')
INSDF1	C	CONST('INSD01')
RNTDF1	C	CONST('RNTD01')
RNTDF2	C	CONST('RNTD02')
RPRDF1	C	CONST('RPRD01')
COMDF1	C	CONST('COMD01')

## Pre-formatted comment text:

COM002	'ARMS reservation transferred from	to Claims
COM003	'ARMS reservation transferred from	to

## Process

Hierarchical numeric ID: 1.1.1.1.6.2.25  
 Coded name: EC00RNV1  
 Name: PGM Generate EC Rental Notification (EC00RNV1)  
 Comment: @Purpose:

To generate an opening or a closing rental notification functional group type, 'RN', transaction data sets for sending to a Rental Management Trading Partner host application system.

## @Operational Method:

- Receive the next DQAM60V1 data queue entry with the key value of 'RCB' in infinite wait mode.

- If the received data queue entry is a shutdown request, send that data queue entry to the DQANDST data queue to inform the AM0106 program that this program's shutdown has been completed and end the program.

- If the received data queue entry is a non-shutdown request with an action code of '3', then attempt to retrieve the record from the RACMAST open rental contract/ticket file first. If not found attempt to retrieve from QRACCLSQ. Populate the associated record formats' fields including loading the Contract Date to Start Date field and Time Out of Rental Segment #1 to the Start Time field, leaving the End Date and End Time fields initialized to zeroes to signify this is an opening rental notification.

- If the received data queue entry is a non-shutdown request with an action code of '4', then retrieve the record from the QRACCLSQ closed rental contract/ticket file and populate the associated record formats' fields including the End Date and End Time fields, leaving the Start Date and Start Time fields initialized to zeroes to signify this is a closing rental notification.

- If the received data queue entry is a non-shutdown request with an action code of '5', then attempt to retrieve the record from the RACMAST open rental contract/ticket file first. If not found attempt to retrieve from QRACCLSQ.

### ARMS Process Report

Populate the associated record formats' fields including the End Date and End Time fields, leaving the Start Date and Start Time fields initialized to zeroes to signify this is a closing rental notification.

- If the received data queue entry is a non-shutdown request with an action code of other than the previously specified values or there was no successful database retrieval, then generate a formatted program dump, call/execute AM0097 program to notify the ARMS On-Call of this failure and reject the transaction for further processing.

- Load the OFFD01V1 format with the passed data queue entry's contract/ticket ID and its location, along with the retrieved shipping name, street address, city, state, postal code and the telephone number for that Enterprise Rent-A-Car rental office.

- Write Rental Notification (RN) group type transaction data set records to transaction data set output file for sending from the distributed rental application system host platform to the ARMS centralized host computer system platform.

- Send the written Rental Notification group type transaction set record formats' key value to the ANDIST output file's associated DQANDST data queue.

- Sends the received input data queue entry back to the DQAM60V1 Dispatch data queue with the key value of 'DIS' as a feedback method to the AM0060V1 dispatch program to indicate that the dispatch task has been completed.

@Files: (CRUD)

- OFFDRB (-R--)
- RACMAST (-R--)
- QRACCLSQ (-R--)
- ANDIST (C---)

@Constants:

'7680' is the sole Group/Branch ID key value used to access its street/city/state address from the Branch Office Directory file (OFFDRB) at the beginning of the program.

8002277800 as the overriding telephone number loaded into all ELCD01:ELCPHN fields.

#### Process

Hierarchical numeric ID: 1.1.1.1.7

Coded name:

Name: AUT Send Transaction for Centralized Processing (AM106 - AM105)

Comment: @Definition: The automatic process of sending the transaction data set from the distributed machine to the centralized machine for processing, via ICF files.

#### Process

Hierarchical numeric ID: 1.1.1.1.7.1

Coded name: AM0106

Name: PGM Send Transaction for Centralized Processing (AM0106)

Comment: @Purpose:

To send transaction data sets from the Distributed machine to the Centralized machine for processing.

ARMS Process Report
---------------------

## @Operational Method:

- Acquire the program device 'AM0106'
- Wait indefinitely for the next DQANDST data queue entry
- IF the receive data queue entry is a shutdown request
  - Increment counter. When the count equals the number of ARMS never ending transaction programs running on that host platform close program files and send that request to ICF device AM0106 to inform the centralized receive program that AM0106 has ended.
  - Submit job for error report and history file update(CELL797).
- ENDIF
- IF the receive data queue entry is a non-shutdown request
  - Retrieve the application transaction data set records
  - Write the data set to ICF device AM0106 to send transaction data set to the centralized ARMS host platform
- ENDIF.

@Notes: This program is currently evoked from the executing Receive Transaction from Distributed Host System (AM0105) program on the centralized ARMS host platform to be the source of the sender linked ICF communication session. Presently, the distributed rental system host platform are all systems excluding BIRELAND, VGERMANY, and UK.

@Files: (CRUD)

- ANDIST (-R--)
- AMVRGTST (-R--)

## Process

Hierarchical numeric ID: 1.1.1.1.7.2

Coded name: AM0105

Name: PGM-Receive Transaction from Distributed Host System (AM0105)

Comment: @Purpose:

To receive transaction data sets from the distributed rental system host platform on the ARMS centralized host platform.

## @Operational Method:

- When starting the program, attempt to acquire the device up to 5 times with a one minute delay between each try before signalling an error.
- Receive input transaction from the distributed rental system host platform
- IF data was received
  - IF transaction data set is equal to Vendor Error(ER) or Vendor Office Information(OF)
    - Write records to pre-package file(AMPACK)
    - Send data queue entry to pre-package (DQAMPKG)
  - ELSE
    - Write records to Edit Transaction file(AMSET)
    - Send data queue entry to Transaction Editor (DQAM25)
- ENDIF

<b>ARMS Process Report</b>
----------------------------

ELSE

- IF shutdown request receive
  - Send data queue entry to Transaction Editor (DQAM25)

ENDIF

- Release Device

ENDIF

@Files: (CRUD)

- AMPACK (C---)
- AMSET (C---)

@Notes: It is currently submitted by the Start-Up ARMS Jobs (CLL810) program on the centralized ARMS host platform to be executed with a single 1-character input parameter of the distributed System (Machine) ID to start linked ICF communication session. Presently, the distributed rental system host platform are all systems excluding BIRELAND, VGERMANY, and UK.

## Process

Hierarchical numeric ID: 1.1.1.1.7.3

Coded name:

Name: DTQ Input to AM106 (DQANDST)

Comment: @Definition: DQANDST is a data queue used to provide input to PGM AM0106 which sends transactions from the distributed machine to the centralized processing machine.

## Process

Hierarchical numeric ID: 1.1.1.1.7.7

Coded name: CLL797

Name: PGM Completed Errors Report (CLL797)

Comment: @Purpose: This program is one of the programs along with AM0797 and AM0099 that were intended to implement Tom Stratton's error handler system for error capture and response.

.. @Operational Method:

- Read files ANDQER and AMERDT to gather the details for any error that was marked as completed in AM0099.

@Notes: These programs are not used at this time even though they are on production systems. Essentially this was an implementation failure due to complexity and user friendliness issues.

## Process

Hierarchical numeric ID: 1.1.1.1.8

Coded name:

Name: AUT Package Transaction into Transmission (AM120)

Comment: @Definition: The automatic process of preparing a transaction for transmission by putting it in a transmission envelope.

## Process

Hierarchical numeric ID: 1.1.1.1.8.1

ARMS Process Report
---------------------

Coded name: AM0120

Name: PGM Package Transaction (AM0120)

Comment: Purpose: To add control start and end record formats to the transaction data sets to create an enveloped transmission. These control formats are used to route the transmission and indicate the type of transaction set being sent.

## @Operational Method:

- Receive the next communication acknowledgement, authorization maintenance request, rental maintenance recognition or authorization maintenance error from data queue DQAMPKG until a shutdown request is received.
- Upon receipt of shutdown requests, pass request to time line DQAM70V1.
- For each transaction data set
  - Build the control start record formats(Transmission, Group & Set) and write record formats to send file(AMSEND)
  - For each transaction data set record format
    - Translate the location(AUTD01/ELCD01/IRBH01)
    - Write record format to send file(AMSEND)
  - Build the control end record formats(Set, Group, & Transmission) and write record formats to send file(AMSEND)
  - Send a data queue to the connect-specific communications sender

## @Files: (CRUD)

- AMPACK (-R--)
- ARMSPR3 (-R--)
- AMXBCO (-R--)
- AMSEND (C---)

## Process

Hierarchical numeric ID: 1.1.1.1.8.6

Coded name: DQAMPKG

Name: DTQ Input for Program AM0120 (DQAMPKG)

Comment: @Definition: DQAMPKG is a data queue used to provide input to PGM AM0120 which packages a trasaction in preparation to sending it out to an EDI Trading Partner.

## Process

Hierarchical numeric ID: 1.1.1.1.9

Coded name:

Name: AUT Dispatch Vendor Authorization Maintenance Request

Comment: @Definition: The automatic process of updating the ARMS database on the distributed machine and dispatching the data queue entry for the appropriate format generator program in response to some action by the Rental System.

## Process

Hierarchical numeric ID: 1.1.1.1.9.1

Coded name: AM0060V1

Name: PGM Dispatch Rental Systems Request (AM0060V1)

Comment: @Purpose:

### ARMS Process Report

To accept requests from the rental systems to maintain the distributed ARMS Rental Transaction Cross-Reference and dispatch transaction set generation requests.

#### @Operational Method:

- Override the printer file outq to ARMSDUMP so that the dump can be retained for as long as needed. The default outq ERRDUMPS is cleared daily.

- Receive authorization management or rental notification requests (AC, IN, RA, RC, RE, Opening RN, Closing RN, TR and VM) from the RENTAL SYSTEMS. All requests or notifications as output from the rental systems with a key of 'DIS' will be read from the input data queue (DQAM60V1) by this program.

- Based on externally described rules (AMPGMTBL, AMPGMRUL and AMPGMCTL), dispatch the request to the appropriate request handler process (format generator program e.g., EC00RAV1) dispatching an output data queue entry that has been modified from the input data queue entry to the correct rental system transaction set generator never-ending program/job, based on the key information found in the AMTRNCTL file (ACB, ACC, RAB, RAC, RCB, RCC, REB, REC, RNB, RNC).

- A transaction set generation request may lead to another request. After each request is sent out, a feedback entry to this program informs that the dispatched request has been completed.

- Edit to check for valid Program ID, Rule ID and dispatch execution condition. Generate an error message for an invalid dispatch execution condition. No page is generated for invalid dispatch execution condition. However, for invalid Program ID or Rule ID, ARMS On-Call pager will be sent an informational message. In all cases, transaction request is rejected.

- Receive and process environment change requests which allow the program to dynamically adapt for load balancing during heavy transmission load (Group Type = 'WT') or to shutdown the format generators and itself.

- Log the received data queue entry for audit and problem resolution.

- Retrieve the information required to process the request via appropriate retrieval module (EC2090V1 for ECARS and CC2090V1 for Claims Connection requests). If the retrieval program detects a lock on the ticket, defer processing the request.

- A request may need to be processed in more than one pass through the program. The current step being processed is logged externally (in AMTRNCTL). All requests related to this transaction id are deferred while a step for the transaction id is in progress. When the logged request has been completely processed, the log for that transaction is removed so that the deferred requests can now be processed.

#### @Files:

ARMSPR1	(-R--)
AMPGMTBL	(-R--)
AMPGMRUL	(-R--)
AMPGMCTL	(-R--)
AMTRNCTL	(CR-D)
AMXBCO	(-R--)
AMVRGTST	(-R--)
AMQUETBL	(-R--)
AMSTSX	(-R--)
AM0060LG	(C---)



<b>ARMS Process Report</b>
----------------------------

```

ARMSKEY      (-R--)
ANDSG1V1     (-R--)
$#EMACH      (-R--)
AMPGMCTL     (-R--)
DQAM60V1     (-R--)
DQAM60V1     (-R--)
  
```

**@Notes:**

- AC (Authorization Confirmation)
- IN (Invoice)
- RA Add (Request to Authorize a reservation or an open ticket)
- RA Change (Request to Change an Authorization for a reservation or an open ticket)
- RC (Request to Cancel an authorization)
- RE (Request to Extend a rental)
- Opening RN (Rental Notification of an OPEN ticket)
- Closing RN (Rental Notification of a CLOSED ticket)
- TR (Request to Transfer Authorization from one location to another)

**Process**

Hierarchical numeric ID: 1.1.1.1.9.10  
 Coded name: DQAM6AV1  
 Name: DTQ Secondary Input to AM0060 (DQAM6AV1)  
 Comment: @Definition: DQAM6AV1 is a data queue used as a backup queue for PGM AM0060V1 when it receives a request that can not be processed due to work in process as represented by a record in file AMTRNCTL. This secondary queue is periodically checked when there has been a set period of inactivity in the primary data queue for PGM AM0060V1.

**Process**

Hierarchical numeric ID: 1.1.1.1.9.12  
 Coded name: DQAM60V1  
 Name: DTQ Primary Input for AM0060 (DQAM60V1)  
 Comment: @Definition: DQAM60V1 is a data queue used to provide the primary source of input to PGM AM0060V1 which handles processing needed as a result of requests and or actions taken by the Rental System (ECARS) and routes requests for formats to be generated to the correct format generator programs.

**Process**

Hierarchical numeric ID: 1.1.1.1.9.13  
 Coded name: EC2090V1  
 Name: PGM Retrieve ECARS Data (EC2090V1)  
 Comment: @Purpose:

To retrieve specified information from the ECARS rental database.

**@Operational Method:**

- If the passed ARMS DS Identification code is 'G1', attempt to retrieve the data otherwise if the code is 'GO', return to calling program.
- If the reservation number is passed, get the reservation data.

### ARMS Process Report

- If the ticket number is passed, get the ticket data.

#### @Files:

- RACBRMAST	(-R--)	
- NR025L01	(-R--)	
- RACMAST	(-R--)	
- RACSMAS	(-R--)	
- QRACCLSQ	(-R--)	
- RACSCLD	(-R--)	
- RACPENDC	(-R--)	
- ANDSG1V1	(-R--)	EDS
- DTPARM	(-R--)	EDS

#### @Notes:

A reservation created by National Reservations has the routed Machine ID as the first character of the RESERVATION ID.

This program is called by a variety of non-ECARS programs. Each program calling this program may have a different parameter list associated with the program call. Each different parameter list, defines a different type of function required by this program.

Functions desired for this program must be closely coordinated with any NON-ECARS rental application development staff. All objects involved with any desired function must be moved into production at the same time.

This program has been designed and written to be as modular as possible. All logic for each type of transaction is contained in a single subroutine. The required files are opened at the beginning of each subroutine, the other files are not opened.

#### @Improvement Opportunities:

1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

2.) Delete the constant 'EC2090V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.

3.) Convert program from OPM RPG to ILE RPG.

#### Process

Hierarchical numeric ID: 1.1.1.1.9.14  
 Coded name: CC2090V1  
 Name: PGM Retrieve Claims Connection Data (CC2090V1)  
 Comment: @Purpose:

To retrieve specified rental information from the Claims Connection database.

#### @Operational Method:

- Retrieve the Claims Connection reservation or ticket data from the CC database and return the information using pre-defined external data structure.

<b>ARMS Process Report</b>
----------------------------

**@Files:**

```

CCMASTER      (-R--)
CCCLSC         (-R--)
CCRES          (-R--)
ANDSG1V1       (-R--) EDS

```

**@Note:** When a ticket is opened, the reservation is deleted from the reservation master file.

**@Improvement Opportunities:**

- 1.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.
- 2.) Delete the constant 'CC2090V1' and replace its usage with the Program Status Data Structure data element for PROGRAM ID.
- 3.) Convert program from OPM RPG to ILE RPG.

**Process**

Hierarchical numeric ID: 1.1.1.1.9.15  
 Coded name: CLDQINFO  
 Name: PGM Retrieve Data Queue Information (CLDQINFO)  
 Comment: @Purpose:

This program receives a parameter and using the Application Program interface (API) QMHQRDQ, retrieves a data queue description.

**@Operational Method:**

- Prepare the parameters used by the API then call the API
- Parse each attribute retrieved:
  - Length (binary)
  - Key length (binary)
  - Sequence type (FIFO, LIFO, KEYED)
  - Sender ID (Y or N)
  - Force Indicators (Y or N)
  - Description
  - Number of messages on the queue (binary)
  - Max. number of message on queue (binary - not used)
- Convert binary values to decimal
- Return to values to calling program.

**Process**

Hierarchical numeric ID: 1.1.1.1.18  
 Coded name:  
 Name: AUT Dispatch Customer Authorization Maintenance Request  
 Comment: @Definition: The automatic process of updating the ARMS database on the distributed machine and dispatching the data queue entry for the appropriate program to then update the Rental database.

**Process**

# ARMS Process Report

Hierarchical numeric ID: 1.1.1.1.18.3  
 Coded name: AM0061V1  
 Name: PGM Edit and Dispatch Customer Request (AM0061V1)  
 Comment: @Purpose:

To perform the distributed Edits and Dispatch a Customer Request transaction to the appropriate Rental System never-ending program/job or Financial System Interface daily batch transaction files.

## @Operational Method:

- Execute as a never ending program on the distributed host machine.
- Receive the next data queue entry with a key value of 'DIS' from data queue DQAM61V1 or DQAM6BV1.
- If a 'WT' (Wait Time) group type is received, change the amount of time the primary data queue must elapse before switching to the secondary data queue to the value in group type. However if the value = 99999, set the wait time to -1, infinite.
- If a 'DF' (Deferred) group type is received, execute the program to return the number of entries in secondary data queue.
- If a 'SD' (Shutdown Request) group type is received, 1 pass is made through the secondary data queue. Any entry not processed will remain deferred. After passing through the secondary data queue, generate shutdown request for the necessary rental interface program(s).
- If an 'RM' (Remittance Advice) group type is received at the distributed host machine that is not the financial systems host, only update the cross-reference status.
- If a daily rental transaction (excluding RM) is received at the distributed host machine that is the financial systems host, the program dumps and invokes error logging process, AM0098.
- If the authorizing customer is different from the original customer on an old style vendor override, generate a request for cancellation for the original customer who received the original request for authorization.
- After business rules have been validated, the request is either rejected back to the trading partner, passed on the appropriate interface request handler process, and in certain cases the request is deferred. (A key comprised of the transmission group type and business type is created when passing the request to the interface program)
- Log the rejection of a request for authorization for an open ticket (in file AMSTX).
- If the ticket referenced in the vendor override request is not found, generate standardized comment stating that the vendor override failed (data area AMATTCDA, format COMD01 in file AMAPP).

## @Files:

AMAPP	(CRU-)
AMVRGTL1	(-R--)
AMQUETBL	(-R--)
AMSTX	(CR--)

<b>ARMS Process Report</b>
----------------------------

**@Notes:**

- AT (Authorization Add/Change/Transfer)
- CC (Customer Confirmation)
- CN (Customer Cancellation Request)
- CM (Customer Message)
- EX (Rental Extension)
- PM (Customer Payment Advice)
- RM (Customer Remittance Advice)

The printer file for error DUMPS QPPGMDMP is overridden in this program to go to outq ARSMDUMP instead of the Enterprise default of ERRDUMPS because the outq ERRDUMPS is cleared on a daily basis and sometimes the ARMS on call staff can not get around to fix the problem under consideration for more than a day. This way the dump is retained until explicitly deleted.

\*\*\*\*\* External Data Structures \*\*\*\*\*

```

ANDSG1V1
ARMSKEY
APPD01V1
CRED01V1
COMD01V1
PMTD01V1
RMTD01V1
DQAM60V1
AMATTCD          (-R--) Data Area
$#EMACH
  
```

**Process**

Hierarchical numeric ID: 1.1.1.1.18.4  
 Coded name:  
 Name: DTQ Secondary (Deferred) Input to AM0061 (DQAM6BV1)  
 Comment: @Definition: DQAM6BV1 is a data queue that is used as a source of deferred input to PGM AM0061V1 which routes Trading Partner transactions to the correct Rental System interface program. This data queue is input to by EC00ATV1 and AM0061V1.

**Process**

Hierarchical numeric ID: 1.1.1.1.18.5  
 Coded name: DQAM61V1  
 Name: DTQ Primary Input to AM0061 and Rental System Interfaces (DQAM61V1)  
 Comment: @Definition: DQAM61V1 is a data queue used to provide input to PGM. AM0061V1 and Rental System Interface programs which process inbound transaction requests from the EDI Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.1.1.18.8  
 Coded name: AM0049V1  
 Name: PGM Update Financial Services Files with Remittance Detail (AM0049V1)  
 Comment: @Purpose:

To write or update records to batch transaction header and detail holding files that are used to update the FINANCIAL SYSTEMS' Cash Receipts application database files with the detailed information of EFT remittance advice detail transaction data sets that were sent and processed through the ARMS system. This

### ARMS Process Report

Information will be used by Financial Systems to process Electronic Funds Transfer and to distribute and apply the incoming funds. There will be one header record for each EFT batch received.

#### @Operational Method:

Using the passed input parameter field values for COMPANY PROFILE ID for Ins.ID, EFT IDENTIFICATION, and a SEQUENCE NBR. starting at 1 through 9 as a key, attempt to access for update an existing associated Cash Receipts EFT Remittance Advice Header File (CREFTH) record.

IF no record is found, write a new record, using this key value. Load the passed input parameter PAYMENT AMOUNT field value to the output BATCH TOTAL field and load GROUP ID with "76" and BRANCH ID with "99" and load the RECORD ADDED audit fields appropriately.

IF a record is found, check its STATUS, being one of the following values:

```
blank = Unprocessed
I      = In use by another user
P      = Pending
S      = Submitted to batch
O      = Processed or Posted
D      = Deleted
```

IF this batch record's STATUS is submitted ("S"), posted ("O"), or, deleted ("D"), then the record is unavailable to ARMS interface. Increment the key's SEQUENCE NBR. by 1 and reattempt update access. However, IF there were 9 records found that all were unavailable, then DUMP and notify ARMS On-Call of the exception error by calling 'AMPSSR' with the passed input parameter field being a concatenation of the passed input parameter fields' values to this program.

ELSE, update this record's accumulated BATCH TOTAL with the passed input parameter field, PAYMENT AMOUNT.

IF no exception error has occurred, then attempt to access for update an existing associated Cash Receipts EFT Remittance Advice Detail File (CREFTD) record, using the same key values last used for the Header File.

IF no record is found, write a new record, using this key value and a LINE NBR value of 1. Load the passed input parameter field, PAYMENT AMOUNT value to the output PAYMENT TOTAL field and load the passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D" and load the RECORD ADDED audit fields appropriately.

IF a record is retrieved, compare the retrieved record's GROUP ID, BRANCH ID, and TICKET ID to the passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D".

IF they match, then add the passed input parameter PAYMENT AMOUNT field value to the current PAYMENT AMOUNT field and update the RECORD.CHANGED audit fields appropriately.

IF no records are found whose GROUP ID, BRANCH ID and TICKET ID match, then write a record to this file using the same key fields' values and set the LINE NBR. to 1 or 1 more than was last retrieved. Load the passed passed GROUP ID, BRANCH ID and TICKET ID prefixed with "D", ENTERPRISE INTERNAL CUSTOMER ID, PAYMENT AMOUNT to their associated fields and load the RECORD ADDED audit fields appropriately.

@Files: (CRUD)

ARMS Process Report
---------------------

CREFTH (CRU-) Cash Receipts EFT Payment Remittance Advice Header File,  
by , by Ins. ID, EFT ID#, and Sequence Nbr.

CREFTD (CRU-) Cash Receipts EFT Payment Remittance Advice Detail File,  
by Ins. ID, EFT ID#, Sequence Nbr., and Line Number.

**@Notes:**

This program is called by the ARMS Distributed Edits and Dispatch  
(AM0061V1) program with the following input parameter fields:

5	Alphanumeric	Company Profile ID
20	Alphanumeric	EFT Identification
2	Alphanumeric	Group ID
2	Alphanumeric	Branch ID
6	Alphanumeric	Rental Contract/TICKET ID
7	Alphanumeric	Enterprise Internal Customer ID
15/5	Numeric	Payment Amount

For any program exception error, the \*PSSR - ERROR HANDLING SUBROUTINE  
will be executed that will output a spooled RPG formatted DUMP and CALL 'AMPSSR' with  
the passed input parameter field being a concatenation of the passed input parameter  
fields' values to this program to notify the ARMS On-Call staff.

The ARMS system will now enable electronic invoices, payments and  
remittances (processed here) to be sent/received for non-ARMS Rental Management  
Trading Partners that are Direct Billing/Remittance Trading Partners, such as  
Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS  
will receive a data queue entry to generate an electronic invoice (and create an  
associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the  
trading partner never sent to ERAC (via ARMS) any initial rental authorization.

**@Embedded Data/Constants:**

'AM0049V1' is used as a literal constant to pass as an input parameter  
field value to the calls of 'AMPSSR' program and the output of the PROGRAM ID audit  
field in the CREFTD and CREFTH files' records.

'7699' (Host) is always used to load the GROUP and BRANCH ID fields of  
the Cash Receipts EFT Payment Remittance Advice Header File

**@Improvement Opportunities:**

Replace the literal constant 'AM0049V1' with the Program Status Data  
Structure's data element for PROGRAM ID in the RECORD ADDED and RECORD CHANGED audit  
fields.

**Process**

Hierarchical numeric ID: 1.1.1.1.19

Coded name:

Name: AUT Send Transmission to Trading Partner Company

Comment: @Definition: The automatic process of translating (if necessary) and  
sending a transmission to the Trading Partner through one of several EDI mediums.

**Process**

Hierarchical numeric ID: 1.1.1.1.19.2

Coded name: EDZSCWA

**ARMS Process Report****Name:** PGM Check for Wrapped Transactions to Send (EDZSCWA)**Comment:** @Purpose:

To check for Outbound wrapped EDI transactions for Any EDI integration software sender processing. These transactions have been wrapped by program EDLSIMA in collaboration with program AMZSIMA.

**@Operational Method:**

This program receives a data queue entry keyed by the trading partner communications port id and if it is not a shutdown entry, it will read the Any EDI integration software Connection Log file to determine if there are any outbound wrapped messages ready to be sent for this trading partner.

Based on the above information the return parameters are set.

**@Files:**

EXLLCPP (in library EXTSSSF) (-R--)

**Process****Hierarchical numeric ID:** 1.1.1.1.19.7**Coded name:** AMSCPIC**Name:** PGM Send Proprietary Transmission to an ARMS-connected VAN (AMSCPIC)**Comment:** @Purpose:

To request a program start communication conversation from an ARMS trading partner, then continuously sends proprietary EDI data set transmissions to the trading partner. These transmissions are triggered by the receipt of their key value via an input data queue entry until a data queue entry of \*DOWN is received from the input data queue.

**@Operational Method:**

For any error exception condition that occurs (including CPI-C errors), output a spooled RPG formatted dump and CALL 'AMPSSR to notify ARMS On-Call staff of the condition and end the program.

**At program startup:**

- Derive ARMS to an ARMS-connected VAN Transmissions to be sent input data queue name by concatenating 'DQ' with input parameter COMPANY PROFILE ID value (example 'DQVN101').

- Call QCMDEXC to execute the ADDLIB QSYS2 command to access the OS/400 System CPI-C objects.

After startup has completed, repetitively perform the following steps:

- Receive the next data queue entry from the derived input data queue into the ARMSKEY external data structure, waiting infinitely for an entry.

- IF after any of the subsequent program executions, the returned PROGRAM RETURN CODE is not equal to 0 (zero), then execute the program exception error routine.

- Initialize a CPI-C communication conversation by executing 'CMINIT' program with the following parameters:



ARMS Process Report			
---------------------	--	--	--

8	Character	(Output)	CONVERSATION ID
5	Character	(Input)	CONVERSATION SYMBOLIC DESTINATION ID

(CSI), loaded with the input parameter COMPANY PROFILE ID.

9/0	Binary	(Output)	PROGRAM RETURN CODE
-----	--------	----------	---------------------

- Allocate a CPI-C communication conversation by executing 'CMALLC' program with the following parameters:

8	Character	(Input)	CONVERSATION ID
9/0	Binary	(Output)	PROGRAM RETURN CODE

- Set a CPI-C communication session/conversation send type by executing 'CMSST' program with the following parameters:

8	Character	(Input)	CONVERSATION ID
9/0	Binary	(Input)	CONVERSATION SEND SET TYPE, loaded with the value 1.
9/0	Binary	(Output)	PROGRAM RETURN CODE

- IF the received data queue entry's COMPANY ID = '\*DOWN', or the GROUP TYPE CODE = 'SD', or the CUSTOMER TRANSACTION ID = '\*STOP', then execute the following two programs, then end this program.

- - The first is to set the CPI-C Deal Type by executing 'CMSDT' program with the following parameters:

8	Character	(Input)	CONVERSATION ID
9/0	Binary	(Input)	CONVERSATION DEAL SET TYPE, loaded with the value 1.
9/0	Binary	(Output)	PROGRAM RETURN CODE

- - The second is to DeAllocate a CPI-C communication conversation by executing 'CMDEAL' program with the following parameters:

8	Character	(Input)	CONVERSATION ID
9/0	Binary	(Output)	PROGRAM RETURN CODE

- IF a non-shutdown data queue entry is received, then use its value to repetitively read each corresponding record in the ARMS Transmissions to be Sent (AMSEND) file. Load each retrieved record's ARMS format to a temporary DATA TO BE SENT array of 128 character elements and 68 elements, increment the RECORD COUNTER by 1 and the SENT DATA LENGTH by 128.

- - IF the RECORD COUNTER is greater than 69 (68 x 128 = 8704 characters, full communication buffer), or at least 1 record was read from the ARMS Transmissions to be Sent (AMSEND) file, then execute the 'CMSEND' program with the following parameters:

8	Character	(Input)	CONVERSATION ID
8704	Character	(Input)	DATA TO BE SENT ARRAY
9/0	Binary	(Input)	SENT DATA LENGTH (loaded with the current increments of 128 bytes per record).
9/0	Binary	(Output)	REQUESTOR ID, loaded with 0 (zero).
9/0	Binary	(Output)	PROGRAM RETURN CODE

- - Read from the loaded DATA TO BE SENT ARRAY, starting at the first element for every element loaded, write a record to the ARMS Transmissions Sent (AMSENDLOG) file and then send a data queue entry for these written records to the ARMS Time Line Input Data Queue (DQAM70V1).

- - Reset the SENT DATA LENGTH and the RECORD COUNTER to 0 (zero) and clear the DATA TO BE SENT ARRAY.

<b>ARMS Process Report</b>
----------------------------

Files:	(CRUD)
--------	--------

AMSEND	(-R--)
AMSNDLOG	(C---)

**@Embedded Data/Constants:**

'ADDLIBLE LIB(QSYS2)' for use in the execution of the program QCMD5XC to execute this command so that the IBM System Library for CPI-C programs' objects would be available for use.

'\*LIBL' used as the input parameter for the LIBRARY NAME for the execution of the Receive Data Queue and Send Data Queue programs.

**Process**

Hierarchical numeric ID: 1.1.1.1.19.8

Coded name: EDMSLUA

Name: PGM Send X12 Transmission (EDMSLUA)

Comment: @Purpose:

To send any wrapped EDI X12 transactions that are ready to be sent.

**@Operational Method:**

- This program runs in an infinite loop from the time it is submitted until a shutdown flag is returned by the called program EDZSCWA. The program is submitted by the ETD: EC/EDI: ARMS Trading Partner Insurance Company Comm./Mapping Submitter (AM\*\*\*COMP) program to run in the ARMS subsystem on the centralized ARMS host computer platform and is passed the following input parameters:

5 Character COMPANY PROFILE ID with transactions to be sent (such as '\*\*\*01')

15 Character NETWORK ID, as defined in Any EDI integration software, for this ARMS Company Profile ID to be used for this network communication session. (such as 'ProfileName')

6 Character PORT ID, as defined in any EDI integration software, to be used for this network communication session (such as 'xxxxx').

10 Character SCRIPT ID, as defined in any EDI integration software, as the communication script to be used for this company in the ARMS connected network. (such as 'xxxxxxS')

1 Character VERSION FLAG, Either '0' = Test or '1' = Production.

- Repetitively execute the ARMS ANY EDI INTEGRATION SOFTWARE Generic Send Transaction Notify Program (EDZSCWA) with the following parameters, until the \*DOWN FLAG indicates a shutdown has been received:

5 Character (input) passed input parameter, COMPANY PROFILE ID with transactions to be sent (such as 'xxxxx')

6 Character (input) passed input parameter, PORT ID, as defined in any EDI integration software, to be used for this network communication session. (such as '\*\*\*01')

1 Character (output) \*DOWN FLAG shutdown data queue entry received. (Values are 'Y' = Yes or 'N' = No)

1 Character (output) TRANSACTIONS PROCESSED TO BE SENT FLAG, (Either 'Y' = Yes or 'N' = No)

1 Character (input) passed input parameter, VERSION FLAG (Either '0' = Test or '1' = Production)

### ARMS Process Report

IF the returned TRANSACTIONS PROCESSED TO BE SENT FLAG is Yes, then Send transactions to trading partner based on Company ID, Network, Port, and Communication's Script by execution of the any EDI integration software's command for API: Send connections ready to send - batch.

- Upon return of control from EDZSCWA, if the shutdown flag has been set, end the program.

- IF there is an error is due to communications error line ARMS-CONNECTED VALUE-ADDED NETWORK - (UEX0567), or,  
 IF there is an error is due to communication's failure, (UEX0573),  
 IF there is an error is due to no connections ready to send which match the selection parameters, (UEX0517), or,  
 IF there is an error while processing a connection, (UEX0519), or,  
 IF there is any other error escape message, then execute the error routine.

The error routine consists of: Output a spooled CL program dump; Notify ETD and ARMS On-Call personnel, end the program.

#### @Notes:

Program EDZSCWA waits on the data queue that indicates the presence of wrapped transactions and returns the indicator for the presence of these transactions to this program.

#### Process

Hierarchical numeric ID: 1.1.1.1.19.12  
 Coded name: EDLSIMA  
 Name: PGM Map Proprietary to X12 Transmission (EDLSIMA)  
 Comment: @Purpose:

To drive the translation of the ARMS proprietary transaction set into the ANSI X12 EDI transmission set.

#### @Operational Method:

IF translation errors are encountered or any unspecified escape error messages are generated, Dump the CL program, Call both the ETD PSSR and the ARMS PSSR programs, passing the current job's JOBNAME, 'EDLSIMA1', and a single blank character to notify ETD and ARMS On-Call staff. Then End the program.

Retrieve the EC/EDI Send Mapper Debug Flag 1 character data area to check the DEBUG FLAG value.

IF the passed input parameter field VERSION is 'T', then perform overrides to the "test" interface files for 272 set to the ELEDITST E's EDI Test Data Library for the following 272 transaction set files and their associated test interface save files: S272DTL1, S272DTP1, S272HDR1, S272ER1, S272K21, S272NAM1, and S272PER1. Additionally for the following 824 transaction set files: S824HDR1, S824ER1, S824OTI1, S824N1, S824NTE1, S824ME1, and S824PER1. Also, IF not on the "DEV" (development) platform, then override the following DDM (distributed data management) remote file on the "DEV" platform in the ELEDITST EDI Test Data Library for the ARMS Cross-Reference File (DAMXREFL1) and the ARMS Authorization Detail File (DAMAUTD).

Allocate the following output sender segment detail interface work files for exclusive use:

ARMS Process Report
---------------------

S272DTL1  
S272DTP1  
S272ER1  
S272HDR1  
S272K21  
S272NAM1  
S272PER1

S824HDR1  
S824ER1  
S824OTI1  
S824N1  
S824NTE1  
S824ME1  
S824PER1

Derive the Trading Partner (COMPANY ID) Send Communications Version 1 output data queue name. IF the passed VERSION is '1' (production), then build the data queue name by the concatenation of the first three (3) characters of the passed input parameter field COMPANY ID, the constant "SE" (to SEnd), and the passed input parameter field VERSION.

ELSE (is test), then build the data queue name by the concatenation of the characters from position 2 through 4 of the passed input parameter field COMPANY ID, the constant "SE" (to SEnd), and the passed input parameter field VERSION.

Perform the following until SHUTDOWN FLAG or an exception error occurs:

- Call the ARMS X12 Send Interface Version 2 program (AMZSIMA) with the following parameters:

1	Character	(Output) Shutdown Flag (possible values: 'N'=No or 'Y'=Yes)
3/0	Decimal	(Output) Number of 272 Transaction Sets
3/0	Decimal	(Output) Number of 824 Transaction Sets
5	Character	(Input) Company ID, passed input parameter field value to this program

- IF transactions were written to the interface files (NUMBER OF 272/824 TRANSACTION SETS > 0), then execute the EDI integration software Create Application Data Batch (batch) command for the Property and Casualty Loss Notification Outbound Version 1 application ID for the production or test data processed, depending on the value of the input parameter VERSION. Then send a '\*DATA' (meaning "data is available to be sent") 5-character data queue entry to the derived output Trading Partner (COMPANY ID) Send Communications Version 1 data queue.

- IF the retrieved data area's DEBUG FLAG is 'Y' (program is running in debug mode), the copy/add the transaction set segment detail interface work files to their associated save archive files.

- Clear these transaction set segment detail interface work files.

- IF the called ARMS X12 Send Interface Version 2 program (AMZSIMA) returned with SHUTDOWN FLAG parameter = 'Y', then send three (3) '\*DOWN' 5-character data queue entries to the derived output Trading Partner (COMPANY ID) Send Communications Version 1 data queue and end this program.

- ELSE, repeat the above steps.

@Files: (CRUD)

ARMS Process Report
---------------------

EDEXTDBUG (-R--) 1 character data area ETD: EC/EDI Send Mapper Debug  
 Flag (possible values: "N"=No or "Y"=Yes)

**@Notes:**

This program is maintained by the Emerging Technologies Department (ETD).  
 This program is submitted as a never-ending program batch job with the  
 job name 'LxxxxxISM' (where "xxxxx" is the Trading Partner Company Profile ID) and  
 specifies using the any EDI integration software JOBDP job description on the  
 centralized ARMS application host computer system with the following input  
 parameters:

1 Alphameric VERSION (possible values: "1"-production or "T"-testing)  
 5 Alphameric COMPANY ID

**Process**

Hierarchical numeric ID: 1.1.1.1.19.30

Coded name: AMZSIMA

Name: PGM Create Send Interface Work Files for Any Transactions to be Sent (AMZSIMA)

Comment: @Purpose:

To receive any outbound transmission to be sent and format it into the  
 appropriate send transaction interface work files for mapping by the EDI integration  
 software (any third-party software package).

**@Operational Method:**

- Set the delay time for gathering transactions from data area AMEXTSM.  
 To improve performance, the EDI integration software is not invoked for each  
 transaction. The delay time is the duration for which the program will wait to see  
 if any more transactions are available to be sent before returning to the calling  
 program.

- Receive data to be sent to insurance company in ARMS proprietary format  
 and convert the format to send interface file format for X12 transaction sets 272  
 {property and casualty loss notification and work assignments (rentals)} and 824  
 {application advice designed to accommodate the business need for reporting the  
 acceptance, rejection, or acceptance with change of any transaction set}.

- Use file EDPREF3 to determine which insurance company expects which X12  
 transaction set for a particular ARMS proprietary group type.

**@Files:           CRUD)**

- AMSEND	(-R--)
- AMTRNLOG	(-R--)
- S272HDR1	(CRU-)
- S272NAM1	(CRU-)
- S272DTL1	(CRU-)
- S272DTP1	(CRU-)
- S272ER1	(CRU-)
- S272K21	(CRU-)
- S272PER1	(CRU-)
- AMSNDLOG	(-R--)
- EDPREF3	(-R--)
- S824HDR1	(CRU-)
- S824ER1	(CRU-)
- S824N1	(CRU-)

ARMS Process Report	
---------------------	--

S824OTT1	(CRU-)
S824PER1	(CRU-)
S824NTE1	(CRU-)
S824MB1	(CRU-)
AMAUTD	(-R--)
AMXREFL1	(-R--)
AMEXTSM	(-R--) Data Area

@Notes: State

Currently, this used for the ANSI X12 '272' Rental Management Transaction Set used for Trading Partner Company and for the ANSI X12 '824' Consolidated Payment Transaction Set used for a different Trading Partner Company.

Currently only one ARMS Trading Partner Company will be sending 824 transactions but CIECA standards will be used in the future by other insurance companies to process status type transactions such as: 'AC', 'RN', 'ER'.

#### Process

Hierarchical numeric ID: 1.1.1.1.20

Coded name:

Name: AUT Generate Authorization Maintenance in Behalf of R.M.T.P.

Comment: @Definition: This automated process allows for an Enterprise In-House person to perform Authorization Management Maintenance on behalf of a Rental Management Trading Partner. Currently this is restricted to the processing of Authorization Changes and Cancellations. Only one ARMS Trading Partner Company is using this currently.

#### Process

Hierarchical numeric ID: 1.1.1.1.20.1

Coded name: AACHNG

Name: PGM AADDRVR Program to PASSTHRU to RARMS (AACHNG)

Comment: @Definition: This program is used to pass through to the Centralized ARMS machine. It is used by In-house Enterprise Rental management personnel to access an interactive screen for use in processing AT-C, and or CN's on behalf of an ARMS Trading Partner Company.

@Operational Method: All this program does is start pass-through and starts the AACHNGA program. End program.

#### Process

Hierarchical numeric ID: 1.1.1.1.20.2

Coded name: AACHNGA

Name: PGM Program to start AACHGR (AACHNGA)

Comment: @Definition: This program is used to facilitate In-house Enterprise Rental management personnel access to an interactive screen for use in processing AT-C, and or CN's on behalf of a ARMS Trading Partner Company.

@Operational Method: This program is started by AACHNG on the Centralized ARMS machine. This program will call program AACHGR and then when control is returned it will end Pass-Through returning the user to the distributed machine.

#### Process

**ARMS Process Rep rt**

Hierarchical numeric ID: 1.1.1.1.20.3

Coded name: AACHGR

Name: PGM ARMS Trading Partner Company Authorization Change (AACHGR)

Comment: @Definition: This program is used to allow In-house Enterprise Rental management personnel to access an interactive screen for use in processing AT-Changes , and/or Cancellations on behalf of a ARMS Trading Partner Company.

@Operational Method: The System Name is retrieved and the users authority to the program is validated. If the user is not authorized, \*INLR is turned \*ON and the program exits.

If the user is authorized, all screen values and indicators are reset and Screen 1 is displayed. F3-Exit will exit the program. F5-Next will reset all screen values and indicators and redisplay the screen. F10-Process will begin processing, if all edits pass, an authorization change or cancellation. F11-Cancel will display Screen 2, if all edits pass.

After the user enters a valid combination of Group, Branch, Reservation, and Ticket, the ARMS cross-reference record is retrieved and the current status is used to determine possible transaction types. Authorization change requests are only allowed when the cross-reference status is 'R', 'O', or 'I'. Cancellation requests are only allowed when the cross-reference status is either 'R' or 'I'.

If an authorization change request is being processed, the new values are compared with the old values to ensure a change is made and edited for validity. If edits pass, necessary formats (APPD01, ADJD01, AUTD01, SURD01, COMD01) are loaded for an AT-C. Records are written to AMSET. A transaction log entry is written to AM004P. Data queue entries are sent to DQAM25V1 and DQAM70V1. Screen 1 is then redisplayed.

If a cancellation request is being processed, necessary formats (APPD01, ADJD01, CAND01, COMD01) are loaded. Records are written to AMSET. A transaction log entry is written to AM004P. Data queue entries are sent to DQAM25V1 and DQAM70V1. Screen 1 is then redisplayed.

@Notes: This is a temporary program for use until the ARMS Trading Partner Company completes development of their system.

@Files: (CRUD)

- AMXREFL1 (-R--)
- AMXREFL2 (-R--)
- AMRNTD (-R--)
- AMADJD (-R--)
- AMAUTD (-R--)
- AMSURD (-R--)
- AMXBCO (-R--)
- CUSTMAST (-R--)
- AMSET (C---)
- AM004P01 (C---)

**Process**

Hierarchical numeric ID: 1.1.1.2

Coded name:

Name: BT Manage Customer Remittance

**ARMS Process Report**

**Comment:** @Definition: The set of automated activities that are used in management of remittances, this is the result of an inbound transmission received from the trading partner who is engaged in electronic invoicing with us.

**@Purpose:** To allow EDI Trading Partners to receive and manage billing invoices electronically.

**Process**

**Hierarchical numeric ID:** 1.1.1.3

**Coded name:**

**Name:** BT Manage Customer Message

**Comment:** @Definition: The set of automated activities that are used in management of customer messages, this is the result of an inbound transmission received from the trading partner who is engaged in electronic messaging with us.

**@Purpose:** To allow the Rental Management Trading Partner to send messages to the Rental location electronically without the need of using the telephone.

**Process**

**Hierarchical numeric ID:** 1.1.2

**Coded name:**

**Name:** AA Rental Systems Business Transactions

**Comment:** @Definition: This application area represents all EDI transmissions that are sent from any of the Enterprise Rent-A-Car current internal rental application systems to the Rental Management Trading Partner system for: authorization requests, notification, cancellations, and extension; Vendor messages; Vendor Invoices

**Process**

**Hierarchical numeric ID:** 1.1.2.1

**Coded name:**

**Name:** BT Request for Authorization Management by Vendor

**Comment:** @Definition: The set of automated activities that are used in the request for authorization and extension of a rental that is seen as an outbound transmission directed toward the trading partner.

**@Purpose:** To provide the rental offices an automated /electronic way to request authorizations and extensions for rentals from Rental Management Trading Partners without having to use the telephone.

**Process**

**Hierarchical numeric ID:** 1.1.2.2

**Coded name:**

**Name:** BT Generate & Send Rental Maintenance Notification

**Comment:** @Definition: The set of automated activities that are used in the creation of and transmission of rental notification seen as an outbound transmission directed toward the trading partner.

**@Purpose:** To notify the trading partner of the open or close of an authorized rental.

**Process**



**ARMS Process Report**

Hierarchical numeric ID: 1.1.2.3

Coded name:

Name: BT Manage Vendor Messages

Comment: @Definition: The set of automated activities that are used in management of vendor messages, this is the result of an action on the part of a rental agent and results in a transmission sent to a trading partner that engages in electronic messaging with us.

@Purpose: To allow the rental office to send electronic messages to the rental trading partner without the need to pick up the telephone.

Process

Hierarchical numeric ID: 1.1.2.4

Coded name:

Name: BT Generate & Send Electronic Billing Invoice and Batch Extensions

Comment: @Definition: The set of automated activities that are used in the creation and transmission of electronic invoices to the trading partner.

@Purpose: To allow for paperless billing of rentals to customers that are set up for EDI billing.

Process

Hierarchical numeric ID: 1.1.2.4.1

Coded name:

Name: BAT Package and Send Transaction / Generate Vendor Extension Request

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to the rental management trading partner or the direct bill trading partner. Or on the distributed rental systems generate requests for authorization extensions (RE's) based upon specific conditions of rentals as determined from the Rental Database

@Purpose: To prepare and send outbound transactions to the EDI Trading Partners. And to insure that rental extensions are requested in a timely fashion.

@Notes: The information held could be invoices, office location updates, service area updates and/or location rate updates.

Process

Hierarchical numeric ID: 1.1.2.4.1.1

Coded name:

Name: BAT Package and Send Transactions to specified ARMS Trading Partner Companies

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to specified ARMS Trading Partner Companies other than ANSI X12.

@Purpose: To prepare and send outbound transactions to the specified ARMS Trading Partner Companies other than ANSI X12 EDI Trading Partners.

@Notes: The information held could be invoices, office location updates, service area updates and/or location rate updates.

Process

ARMS Process Report
---------------------

Hierarchical numeric ID: 1.1.2.4.1.1.5  
 Coded name:  
 Name: DTQ Connect-Specific Communications Sender Input (DQxxxxxx)  
 Comment: @Definition: DQxxxxxx Connect Specific Communications Sender data queues are used to provide input to the programs that send transmissions to our EDI Trading Partners.

## Process

Hierarchical numeric ID: 1.1.2.4.1.1.15  
 Coded name: AM0150  
 Name: PGM Package Batch Transaction (AM0150)  
 Comment: @Purpose:

To package control start records before and control end records after each Vendor Invoice (IN) and each Vendor Extension Request (RE) data set, to form a complete transmission set. Control start and end records are used to route the transmission and indicates the type of transaction set being sent to the Trading Partner.

## @Operational Method:

- For each transaction data set
- Build the control start record format (Transmission, Group, & Set) and write record formats to send file (AMSEND)
- For each transaction data set record format
- Translate the location (AUTD01/ELCD01/IEBH01
- IF record format NE to Insured Detail (INSD01)
- Write record format to send file (AMSEND)
- IF transaction data set EQUALS Vendor Invoice (IN)
- Write record format to ARMS Billing Log
- Build the control end record format (Set, Group, & Transmission) and write record formats to send file (AMSEND)
- Sends a data queue to the connect-specific communications sender

## @Notes:

- Control start/end records must match:
- Transmission start
- Group Start
- Set Start
- :
- Set End
- Group End
- Transmission End

@Files (CRUD)

LS11C (-R--)  
 ARMSPR3 (-R--)  
 AMSEND (C---)  
 AMBILLOG (C---)

## Process

ARMS Process Report
---------------------

Hierarchical numeric ID: 1.1.2.4.1.2

Coded name:

Name: BAT Package and Send Transactions for X12

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to the X.12 trading partner.

@Purpose: To prepare and send outbound transactions to the X-12 EDI Trading Partners.

#### Process

Hierarchical numeric ID: 1.1.2.4.1.2.1

Coded name: UT\_999M

Name: PGM Generic ILE Error Handler (ErrHandler)

Comment: @Purpose:

To handle exceptions for application programs, by determining the proper response reacting in a predetermined way. These responses include sending a MS01 message and sending pages to the appropriate personnel.

@Operational Method:

Applies To: (ILE) Toolbox Error Handling

This generic application program error handler may be executed by an OPM (Original Program Model) program by executing an intermediate ILE RPG wrapper program, "UT\_999M" - ISS:IS Global Error Handling Module (OPM Callable). This program was created with the bound service program "ISLIB/UT\_H05S" - ISS: Universal Toolbox (\*CALLER) that was created with the bound module "ISLIB/UT\_H05S012" that perform the necessary service functions. This program is executed with the following parameters:

	Parameter	Type-Usage	Size	Description
error	programID	Char - Input	10	Program issuing the
	errorID	Char - Input	7	Error ID
	msgDta	Char- Input	75	Substitution
	Return Code	Char - Output	1	Returns 1 for
				failure, 0 for success

Before using this function/procedure, three files must be updated with the necessary information. These files are UT991P, UT992P, and UT993P.

The existing entries must be used as examples to follow.

Syntax:

```
rc = ErrHandler( <programID> :<errorID> [:<msgDta>] [:<errorNoAction>] )
```

Elements:

	Argument	Type	Size	Description
success	rc (Return Code)	Char	1	Returns 1 for failure, 0 for
	programID	Char	10	Program issuing the error

<b>ARMS Process Report</b>
----------------------------

errorID	Char	7	Error ID found in the
UT991P file for the associated Program ID			
msgDta	Char	32767	Substitution data for message
description			
errorNoAction	Char	1	Parameter for future update,
currently has no function			

**@Notes:**

This ILE RPG wrapper program is currently executed only by the AMZ811A program.

The following necessary records were placed into the UT991P ISS:IS Error Message ID File in order to execute the error handling functions properly:

Program Issuing Error	Message Error ID	Message File	Name	Description ID
AMZ811A	NORECFD	AMMSGF	AMM0003	
AMZ811A	SEQERR	AMMSGF	AMM0001	
AMZ811A	INVCMP	AMMSGF	AMM0002	
AMZ811A	*PSSR	AMMSGF	AMM0004	

The following necessary records were placed into the UT992P ISS:IS Error Message Response File in order to execute the error handling response functions properly:

Program Issuing Error	Error ID	Sequence ID	Code	Error Response
AMZ811A	*PSSR	1	ARMSPAGE	
AMZ811A	*PSSR	2	ARMSMS01	
AMZ811A	INVCMP	1	ARMSPAGE	
AMZ811A	INVCMP	2	ARMSMS01	
AMZ811A	NORECFD	1	ARMSPAGE	
AMZ811A	NORECFD	2	ARMSMS01	
AMZ811A	SEQERR	1	ARMSPAGE	
AMZ811A	SEQERR	2	ARMSMS01	

The following necessary records were placed into the UT993P ISS:IS Response/Program File in order to specify the execution of the error handling response mode functions:

Response Code	Exit Program
ARMSPAGE	UT_999M005
ARMSMS01	UT_999M006

**Process**

Hierarchical numeric ID: 1.1.2.4.1.2.19  
 Coded name: AMZ832A  
 Name: PGM Send Interface X12 Transaction Set 832 (AMZ832A)  
 Comment: Purpose:

# ARMS Process Report

To convert Enterprise proprietary EDI formats held on the Centralized ARMS System into X.12 standard formats (transaction set 832) which will then be sent to the X.12 trading partner

## @Operational Method:

- IF transaction have been written to application database (AMSNDLOG), clear application work files.
- Attempt to receive the next DQ832S data queue entry with a key of Company Id.
- IF a data queue entry was not received, close program files (if applicable) and end program
- IF a data queue entry was received THEN
  - IF no records were retrieved from the application database send file(AMSEND), using the received data queue entry - page the on-call staff and get next data queue entry.
  - IF records were retrieved from the application database send file (AMSEND), using the received data queue entry THEN
    - IF the transaction is in the correct sequence, notify the on-call staff
    - Use the Rental Management Trading Partner electronic commerce profile to determine if they receive X12 standard format - 832 for this specific application group type. IF not exit the program.
    - For each invoice, attempt to load the EDI integration software work files:
      - Retrieve records from application database
      - IF the invoice charge code or unit of measure is not found, skip this invoice and notify ARMS on-call.
      - Load the 832 extol work file and writing them to the send log file (AMSNDLOG).
      - Update 832 header work file.
      - Send a data queue entry to timeline (DQAM70)

@Files:	(CRUD)
AMSEND	(-R--)
AMSNDLOG	(CR-D)
AM811P01	(-R--)
S832HDR1	(CR-D)
S832LIN	(CR-D)
S832PER	(CR-D)
S832CTP	(CR-D)
S832N1	(CR-D)
EDPREF3	(-R--)
AM832ERR	(C---)

## Process

Hierarchical numeric ID: 1.1.2.4.1.2.20

Coded name:

Name: DTQ Input for X12 Transaction Set 832 (DQ832S)

Comment: @Definition: DQ832S is a data queue used to provide input for X.12 transaction set 832 to PGM AMZ832A.

## Process

<b>ARMS Process Report</b>
----------------------------

Hierarchical numeric ID: 1.1.2.4.1.2.23  
 Coded name: AMZ811A  
 Name: PGM Send Interface X12 Transaction Set 811 (AMZ811A)  
 Comment: @Purpose:

To convert Enterprise proprietary EDI formats held on the centralized ARMS application system host platform into X12 standard format (transaction set 811) which will then be sent to the X12 trading partner.

@Operational Method:

- IF the transaction set has been written to the application database for Sent Transaction Log (AMSNDLOG) file, then clear the application work files.

- Attempt to receive the next DQ811S data queue entry with a key of Trading Partner Company ID.

- IF a data queue entry was not received, close program files (if applicable) and end program

- IF a data queue entry was received THEN

- IF no records were retrieved from the application database send file(AMSEND), using the received data queue entry - page the on-call staff and get next data queue entry.

- IF records were retrieved from the application database send (AMSEND) file, using the received data queue entry, THEN:

- IF the transaction is in the correct sequence, notify the on-call staff

- Use the Rental Management Trading Partner electronic commerce profile to determine if they receive X12 standard format - 811 for this specific application group type. IF not, then exit the program.

- For each invoice, attempt to load the Extol work files:

-- Retrieve records from application database

-- IF the invoice charge code or unit of measure is not found, skip this invoice and notify ARMS on-call.

-- Load the 811 Extol work file and writing them to the send log (AMSNDLOG) file.

-- Update 811 header work file.

-- Send a data queue entry to timeline (DQAM70)

@Notes:

- This program run with in CLL811 between 3:00am and 5:00am Central time zone.

- This program only processes invoice transmission sets.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer).

@Files: (CRUD)

AMSEND (-R--)

AMSNDLOG (CR-D)

AM801P01 (-R--)

AM802P01 (-R--)

AM811P01 (-R--)

<b>ARMS Process Report</b>
----------------------------

ARMSPR1	(-R--)
AMAUTD	(-R--)
S811HDR1	(CR-D)
S811NAM1	(CR-D)
S811DTL1	(CR-D)
S811IT1	(CR-D)
S811VEH1	(CR-D)
EDPREF3	(-R--)
AM811ERR	(C---)

**Process**

Hierarchical numeric ID: 1.1.2.4.1.2.24

Coded name:

Name: DTQ Input for X12 Transaction Set 811 (DQ811S)

Comment: @Definition: DQ811S is a data queue used to provide input for X.12 transaction set 811 to PGM AMZ811A.

**Process**

Hierarchical numeric ID: 1.1.2.4.1.2.26

Coded name: EDMSOFA

Name: PGM Create Extol Envelope for Office Data (EDMSOFA)

Comment: @Purpose:

To create the transmission envelope for rental location/rates maintenance transaction sets for the sender interface to the Extol EDI Translation Software.

@Operational Method:

- Call this program after the interface program AMZ832A has already put the data in AMSEND. This program will create the necessary EDI transmission envelope and transmit it to the insurance company.

@Notes: Set Environment as ADDLIBLE ELEDIxxx where "xxx" is the first 3 characters of the TRADING PARTNER PROFILE ID.

This program is executed only on the centralized ARMS host application system platform.

This program is currently stewarded by the Emerging Technologies Department's (ETD) Electronic Data Interchange (EDI) projects team.

**Process**

Hierarchical numeric ID: 1.1.2.4.1.2.27

Coded name: EDMSINA

Name: PGM Create Extol Envelope for Invoice Data (EDMSINA)

Comment: @Purpose:

To create the transmission envelope for rental billing invoice transaction sets for the sender interface to the Extol EDI Translation Software.

@Operational Method:

- Call this program after the interface program AMZ811A has already put the data in AMSEND. This program will create the necessary EDI transmission envelope and transmit it to the insurance company.

ARMS Process Report
---------------------

@Notes: Set Environment as ADDLIBLE ELEDIxxx where "xxx" is the first 3 characters of TRADING PARTNER PROFILE ID.

This program is executed only on the centralized ARMS host application system platform and is executed from the ARMS Packaging Translation Job (CLL811) program.

This program is currently stewarded by the Emerging Technologies Department's (ETD) Electronic Data Interchange (EDI) projects team.

#### Process

Hierarchical numeric ID: 1.1.2.4.1.2.28

Coded name:

Name: DTQ Input for X12 Sender (DQxxxSC1) "xxx" = Company

Comment: @Definition: DQxxxSC is a data queue used to provide input for X.12 transaction sender (xxx = Company).

#### Process

Hierarchical numeric ID: 1.1.2.4.1.3

Coded name:

Name: BAT Package and Send Transactions to specified ARMS Trading Partner Companies

Comment: @Definition: Package information held on the Centralized ARMS System into a batch package and send the information to specified ARMS Trading Partner Companies.

@Purpose: To prepare and send outbound transactions to specified ARMS Trading Partner Companies.

#### Process

Hierarchical numeric ID: 1.1.2.4.1.3.7

Coded name: AM\*\*\*SNB

Name: PGM Send Batch Transmission (via ARMS-connected Value-Added Network AM\*\*\*SNB)

Comment: @Purpose:

To send from ARMS to specified ARMS Trading Partner Companies a Batch EDI Transmission of proprietary transaction data sets via ARMS-connected Value-Added Network. The transaction set functional group types sent are rental billing invoice documents ("IN") and rental location/rates maintenance documents ("OF").

#### @Operational Method:

Clear the EDI integration software Local Data Area data structure.

Retrieve the EDI integration software User Profile File record for ACCOUNT = 'ERAC' and USERID = 'ERACARM'.

Load the entire retrieved record value to the EDI integration software Local Data Area data structure's subfield for PROFILE RECORD.

Load the remainder of the EDI integration software Local Data Area data structure's subfields as follows:

'IERACARM '	User files library (E4LBLB)
'IBSTRM '	Stream file name (E4LBST)
'IERCVD '	Received file name (E4LBRC)



### ARMS Process Report

'IEMSGS'	Messages File name (E4LBMS)
'IEFILE'	Received files file name (E4LBFL)
'IESLOG'	Session log file name (E4LBLG)
'0'	Base submit flag (E4LSBF)
'0'	Base calling code (E4LBCL)

IF the passed input COMPANY PROFILE ID parameter is a specified ARMS Trading Partner Companies production ('\*\*\*01'), load the COMMAND FILE NAME with 'IE\*\*\*SND'.

IF the passed input COMPANY PROFILE ID parameter is a specified ARMS Trading Partner Companies testing ('T\*\*\*T'), load the COMMAND FILE NAME with 'IE\*\*\*SNDT'.

Output to this program's Local Data Area (to be used by the EDI integration software) from the loaded EDI integration software Local Data Area data structure.

Execute the EDI integration software base ('IEIFEXEC') program to execute the receive file according to the information just loaded in the current batch job's Local Data Area.

Retrieve the EDI integration software Local Data Area and load into the EDI integration software data structure.

IF the EDI integration software Local Data Area data structure's Base Return Code is not a normal completion ('00'), then execute the Program Status Subroutine (\*PSSR) to produce a formatted RPG dump and execute Perform Internal Error Paging and Messaging ('AMPSSR') program.

Retrieve the current time and date and load into the appropriate Send Log File record fields.

Position the ARMS specified ARMS Trading Partner Companies Send Batch file at beginning of file and then read each record and perform the following until end of file:

..IF the retrieved record's Record Format ID is Transmission Start (TSMS01), move the retrieved record to the Transmission Start record format's data structure. Move the Transmission Start record format's TRADING PARTNER ID and TRANSMISSION CONTROL ID to the appropriate Send Log File fields and to the appropriate ARMS Time Line Data structure subfields.

..Write each retrieved record to the ARMS Send Log File.

..IF the retrieved record's Record Format ID is Transmission End (TSME01), then send a data queue entry to the ARMS Time Line data queue (DQAM70V1) that is associated with the send details written out to the ARMS Send Log.

@Notes:

This program is executed by the CLL811 CL batch program and is executed on the centralized ARMS host system platform (\*RARS\*) in the ARMS subsystem. It is passed a single input parameter of a 5-character COMPANY PROFILE ID.

@Files: (CRUD)

AM\*\*\*SNB (-R--) ARMS Transactions to be Sent to specified ARMS Trading Partner Companies via Send Batch

# ARMS Process Report

IEPROFL (-R--) EDI integration software User Profile File (in the IBM EDI integration software Production Library for \*USER)  
 AMSNDLOG (C---) ARMS Transmissions Sent Log File

Files used as externally-defined data structures:

External Definition  
 EX40LD EDS ETD: EDI integration software Local Data Area  
 TSMS01V1 EDS Transmission Start Record Format  
 DQAM70V1 EDS ARMS Time Line Input Data Queue external description

@Embedded Data/Constants:

'AM\*\*\*SNB' PROGRAM ID

Used as output to the job's Local Data Area for the execution of the EDI integration software base ('IEIFEXEC') program to execute the receive file:

'IEERACARM ' User files library  
 'IESTRM ' Stream file name  
 'IERCVD ' Received file name  
 'IEMSGS ' Messages file name  
 'IEFILE ' Received files file name  
 'IESLOG ' Session log file name  
 '0' Base submit flag  
 '0' Base calling code  
 'IE\*\*\*SND a ARMS Trading Partner Companies production ('\*\*\*01')  
 COMMAND FILE NAME  
 'IE\*\*\*SNDT a ARMS Trading Partner Companies testing ('T\*\*\*T')  
 COMMAND FILE NAME

@Improvement Opportunities:

- 1.) Convert from OPM RPG to ILE RPG program.
- 2.) Replace the constant 'AM\*\*\*SNB' with the Program Status Data Structure subfield PROGRAM ID.

## Process

Hierarchical numeric ID: 1.1.2.4.1.3.10  
 Coded name: AM0125  
 Name: PGM Create Customer Batch Time Work File (AM0125)  
 Comment: @Purpose:

To load a temporary job work file with each of the financial management or rental management trading partners preferred batch packaging/sending start times for production and test transaction data sets.

@Operational Method:

- Using the billing times from the ARMS profile and whether a company receives electronic bills, create the billing times work file for each company id in the ARMS profile (ARMSPR3). If the company is on a VAN (Value Added Network), get the billing time from the corresponding VAN profile.

@Files:

<b>ARMS Process Report</b>
----------------------------

ARMSPR1 (-R--)

ARMSPR3 (-R--)

L811D (C---) ARMS Company Profile IDs and Batch

Packaging/Sending Time preferences job work file, unsorted.

**@Notes:**

The following Trading Partner (Company) Profile ID values are specific ARMS communication connections that multiple trading partner utilize to exchange information with ARMS:

VN101 - an ARMS-connected VAN Information Systems Value-Added Network (VAN) software

VN201 - PC-ARMS

VN301 - ARMS/400

This is executed by the ARMS Batch Packaging/Send Job (CLL811) program on the centralized ARMS application system platform.

The ARMS system also enables electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) even though the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

**@Embedded Data/Constants:**

'AM0125' as PROGRAM ID input parameter for execution of the ARMS Program Status Subroutine Exception Handler (AMPSSR) program.

**@Improvement Opportunitites:**

1.) Convert this OPM RPG to ILE RPG.

2.) Replace the constant 'AM0125' with the Program Status Data Structure's subfield defined for PROGRAM ID.

**Process**

Hierarchical numeric ID: 1.1.2.4.1.3.13

Coded name: AM0140

Name: PGM Create Batch Transactions Work File (AM0140)

Comment: @Purpose:

To copy batch transaction data sets to a job file for batch packaging and sending for a specified trading partner.

**@Operational Method:**

- Write the \*EOBD (End of Batch Data) APPD02 blank data record format to the AMPACK file for the current input parameter TRADING PARTNER PROFILE ID as the COMPANY PROFILE ID portion of the key and the '\*EOBD' as the key's CUSTOMER TRANSACTION ID.

- Retrieve all the data ready to be packaged and sent for a specified insurance company (until the End of Batch Data) is retrieved, written to a job work file, except the End of Batch Data record.

# ARMS Process Report

@Notes: This is executed by the ARMS Batch Packaging/Send Job (CLL811) program on the centralized ARMS application system platform with one 5-character input TRADING PARTNER PROFILE ID.

## @Files:

AMPACK (CR--) Transaction Data Sets to be packaged for sending  
 L811B (C--) Batch Packaging Job Work File  
 ARMSKEY (-R--) EDS  
 APPD01V1 (-R--) EDS

## Process

Hierarchical numeric ID: 1.1.2.4.1.3.14

Coded name: AM0151

Name: PGM Package Batch Transaction for specified proprietary and ANSI X12 ARMS EDI Trading Partner Companies (AM0151)

Comment: @Purpose:

To package control start records before and control end records after Vendor Invoices (IN) and Vendor Extension Request (RE) data set records, to form a complete transmission set. Control start and end records are used to route the transmission and indicates the type of transaction set being sent.

## @Operational Method:

- For each transaction data set  
 --- Build the control start record formats (Transmission, Group, & Set)  
 --- IF company ID EQUALS '\*\*\*01'  
 Write record formats to specified ARMS Trading Partner

Companies send Batch

(AM\*\*\*SNB)

ELSE

Write record formats to send file (AMSEND)

--- For each transaction data set record format

----- Translate the location (AUTD01/ELCD01/IEBH01)

----- IF company ID EQUALS '\*\*\*01'

Write record format to ARMS Trading Partner Companies send

Batch

(AM\*\*\*SNB)

ELSE

Write record format to send file (AMSEND)

----- IF transaction data set EQUAL Vendor Invoice (IN)

Write record format to ARMS Billing Log

--- Build the control end record formats (Set, Group, & Transmission)

--- IF company ID EQUALS '\*\*\*01'

Write record formats to an ARMS Trading Partner Companies send

Batch

(AM\*\*\*SNB)

ELSE

Write record formats to send file (AMSEND)

--- Sends a data queue to the connect-specific communications sender

@Notes:

: ARMS Process Report
-----------------------

```

- Control start/end records must match:
--- Transmission start
    Group Start (IN)
        Set Start
        :
        Set End
        Set Start
        :
        Set End
    Group End(IN)
Transmission End

```

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

```

@Files          (CRUD)

L811C            (-R--)
AM***SNB        (C---)
AMSEND          (C---)
AMBILLOG        (C---)

```

#### Process

Hierarchical numeric ID: 1.1.2.4.1.4  
 Coded name:  
 Name: BAT Generate Vendor Request for Authorization Extension  
 Comment: @Definition: To generate requests for authorization extensions (RE's) based upon specific conditons of the rentals as determined from the Rental Database. This process is triggered by TIME.

@Purpose: To insure that rental extensions are requested in a timely fashion independantly of the manual call back process of the branch without the need of calling a bodyshop or adjustor, for those Trading Partners that allow the receipt of RE's.

#### Process

Hierarchical numeric ID: 1.1.2.4.1.4.8  
 Coded name: CLL811  
 Name: PGM Transmit Batch Transactions (CLL811)  
 Comment: @Purpose:

To perform certain requests/services for our trading partners base on the host platform.

When on the Centralized ARMS system, to package any held information(invoices, office location) into batches and send the transmission to the trading partner.

When on the Distributed Rental Management system, to generate requests for authorization extension.

## ARMS Process Report

### @Operational Method:

- IF the host platform is the distributed rental system (includes claims connection system on CENTRAL), execute the procedure for generating authorization extension request(s).

- IF the host platform is the centralized ARMS system, batch package and send transmission to the trading partner.

1.) Use the Rental Management Trading Partner Profile - billing time, to initiate the batch package process at the predetermined time.

2.) At each scheduled time batch package any held information:

a) Use the Rental Management Trading Partner Profile to set the Trading Partner Company Profile ID.

b) IF Company ID flagged for production, retrieve company specific records from application database(OPNQRYF).

c) IF Company ID flagged as test  
     - IF DEV host platform not available  
         Send error message  
     ELSE  
         Retrieve company specific records from application database on DEV host platform.  
     ENDIF

d) Copy Vendor invoices(IN) and office information (OF) to job file and insert 'bookmark' in application database file.

e) For each company id determine if there are records to package and how to package them:  
     - Single invoice per transmission (ANY ARMS-CONNECTED VAN )  
     - Multiple invoices per transmission  
     - IF Company ID = ARMS Trading Partner Company then EDI format is proprietary  
     ELSE  
         Convert proprietary format to X12  
     ENDIF

3.) Reorganize the EDI integration software User Profile file when the file contains at least 10% of deleted records.

@Files: (CRUD)

L811E	(CR-D)
L811D	(-R-D)
L811C	(C--D)
L811B	(C--D)
AMPACK	(-R--)
AM***SNB	(-R-D)
AM***SNBSV	(C--)

### Process

Hierarchical numeric ID: 1.1.2.4.1.4.9

Coded name: CCOBREV1

Name: PGM Send Claims Connection Vendor Extension Requests (CCOBREV1)

<b>ARMS Process Report</b>
----------------------------

Comment: @Purpose:

For those Rental Management Trading Partners that want this service, to interrogate the extension control file and determine if an authorization extension is needed. A data queue entry will be created for all contracts that have expired.

@Operational Method:

- For each extension control record
  - Verify the contract is authorized with the Rental Management Trading Partner having bill-to responsibility.
  - Determine if the extension / Due Back date has sired and the Rental Management Trading Partner hasn't sent a termination date.
  - Verify the Rental Management Trading Partner allows / is setup for this service - Generate request for extension.

@Files (CRUD)

CCEXTCTL	(-R--)
CCMASTER	(-R--)
REQSTCTL	(-R--)

@Notes:

- Batch jobs runs during the startup of ARMS (L811) between 3:00am and 5:00am cst.
- This program only runs on the centralized financial system(Central).

Process

Hierarchical numeric ID: 1.1.2.4.1.4.16  
 Coded name: ECOBREV1  
 Name: PGM Send ECARS Vendor Extension Requests (ECOBREV1)  
 Comment: @Purpose:

For those Rental Management Trading Partners that want this service, to interrogate the callback control file and determine if an authorization extension is needed. A data queue entry will be created for all contracts that have expired.

@Operational Method:

- For each callback control record
  - Verify the callback is an authorized open ticket with the Rental Management Trading Partner having bill-to responsibility.
  - Determine if the extension date has expired and the Rental Management Trading Partner hasn't sent a termination date.
  - Verify the Rental Management Trading Partner allows / is setup for this service - Generate request for extension.

@Files (CRUD)

CB007P00	(-R--)
RACSMAS	(-R--)
ECEXTCTL	(-R--)
RACSUSP	(-R--)

@Notes:

- Batch jobs runs during the startup of ARMS (L811) between 3:00am and 5:00am cst.

<b>ARMS Process Report</b>
----------------------------

**Process**

Hierarchical numeric ID: 1.1.2.4.2  
 Coded name:  
 Name: AUT Generate CC Rental Billing Invoice  
 Comment: @Definition: The automatic process of generating billing invoices for rentals from third parties via CLAIMS CONNECTION.

**Process**

Hierarchical numeric ID: 1.1.2.4.2.1  
 Coded name: CC00INV1  
 Name: PGM Generate Electronic Invoice Detail (CC00INV1)  
 Comment: @Purpose:

To generate Vendor Invoice(IN) data set for sending to the Rental Management Trading Partner in response to the Claims Connection System closing the rental contract.

**@Operational Method**

- Wait indefinitely for the next DQAM60V1 data queue entry with a key of 'INC'.

- IF the receive data queue entry is a shutdown request, close program files and send that request to data queue DQANDST to inform the distributed send program it has ended.

- IF the receive data queue entry is a non-shutdown request THEN  
   - Rental contract meets authorization criteria  
     - IF the necessary data base records are retrieved THEN  
       - Build the applicable record format(s) and write the

record

to the distribution transaction file(ANDIST). Only

build

RATD01 record format if trading partner profile

indicates it.

ENDIF

- Send a data queue entry on to the Distributed Send to Centralized data queue (DQANDST).

ENDIF

- Send a data queue entry to Dispatch Data Queue (DQAM60V1) with a key of 'DIS' to finish the request.

ENDIF

**@Constants/Embedded Data:**

- DAILY RENTAL - constant  
 - SALES TAX - constant  
 - MISC EXP - constant  
 - SURCHARGE - constant  
 - 43-0724835 (tx id) - constant

**@Improvement Opportunity**

- Use program status data structure when referring to program name.

@Files (CRUD)  
 CCCLSC (-R--)



**ARMS Process Report**

CCSURDTL (-R--)  
ANDIST (C---)  
OFFDRB (-R--)

**@Notes**

- This program only runs on the centralized financial host platform.
- Application Data Set Formats
  - APPD01 - Enterprise internal format
  - IEBH01 - Invoice header format
  - RATD01 - Rate detail format
  - RNTD01 - Renter detail 1 format
  - REND01 - Rental Notification format
  - INSD01 - Insured format
  - IEBD01 - Invoice detail format
  - IEBT01 - Invoice total format
  - ELCD01 - Arms vendor location

**Process**

Hierarchical numeric ID: 1.1.2.4.3

Coded name:

Name: AUT Edit Outbound Transaction

Comment: @Definition: The automatic process of validating or editing the invoice formats and also updating the distributed ARMS database.

**Process**

Hierarchical numeric ID: 1.1.2.4.3.1

Coded name: AM0062V1

Name: PGM Edit Outbound Transaction (AM0062V1)

Comment: @Purpose:

To perform the distributed edit processing on any generated proprietary EDI transaction sets in the ARMS Distributed Transactions Sets to be Centralized (ANDIST) file prior to being sent to the centralized ARMS application host platform for routing.

**@Operational Method:**

This program endlessly receives the non-keyed input data queue entries from the input data queue (DQAM62V1) that the ARMS Dispatch Rental Systems Request program (AM0060V1) generated as input to this program.

Once a shutdown data queue entry is received, then send this shutdown data queue entry to the DQANDST data queue and end this program.

For any non-shutdown data queue entry, check the non-shutdown key Group Type Code to be one of the following values:

'AC' (Rental Transaction Authorization Confirmation)  
'ER' (Rejected Transaction in Error)  
'IN' (Completed Rental Billing Invoice)  
'OF' (Rental Office Location/Rates Maintenance)  
'RA' (Rental Transaction Request for Authorization)  
'RC' (Rental Transaction Request for Cancellation of Authorization)  
'RE' (Rental Transaction Request for Authorization Extension)  
'RN' (Rental Transaction Rental Notification Start/End)

ARMS Process Report
---------------------

IF the data queue entry's Group Type Code is NOT one of the above, then error the transaction set with the ERROR CODE = '16' ("Invalid Group Type Code").

Read each of the input data queue entry's associated Transaction Sets File records (currently only AMINTR is processed).

IF any record format read is not valid for the validated Group Type, then error the transaction set with ERROR CODE = '70' ("Invalid Format Specified in Set")

IF any following Record Format ID's are read, check:

..IF its associated record format data structure has already been loaded, then error the transaction set with ERROR CODE = '71' ("Only One Format of This Type Allowed in Set"):

'ADJD01'  
'ADJD02'  
'APPD01'  
'AUTD01'  
'CAND01'  
'ELCD01'  
'ERRD01'  
'IEBH01'  
'IEBT01'  
'INSD01'  
'OFFD01'  
'REND01'  
'RNTD01'  
'RNTD02'  
'RPRD01'  
'VEDD01'

..ELSE, load the read record format into the record format's data structure.

..IF more than two (2) 'RATD01' record formats are read, then error the transaction set with ERROR CODE = '90' ("Invalid Format - Not Agreed Upon").

..Only allow loading and output of the following maximum occurrences of these Record Format ID's per transaction set, however, do not error the transaction if more are found, just do not allow loading, editing and output of them:

10 'COMD01' Comment Detail  
35 'IEBD01' Invoice Detail  
30 'SURD01' Approved Surcharge Detail

IF GROUP TYPE CODE = 'IN' (Invoice), then do the following:

..Validate that the required formats for this Group Type Code exist ('APPD01', 'ELCD01', 'IEBH01', 'IEBT01', 'INSD01', 'REND01', 'RNTD01'; at least one 'IEBD01'), and IF the 'IEBT01' record format's TOTAL AMOUNT DUE (IETDUE) is not zero. ELSE, error the transaction set with ERROR CODE = '69' ("Required Format Missing from Set").

..Perform transaction set group type specific record format's data element (field) value edits.

..IF no errors are found, then:

### ARMS Process Report

....IF the data queue entry's VENDOR TRANSACTION ID is blank, then do the following (This is for any Financial Management Trading Partner who is able to receive any rental transaction's EDI Invoices but is not a Rental Management Trading Partner - Non-ARMS participating customer) :

.....Add a new ARMS Rental Transaction Cross-Reference File record by executing the ARMS Assign Next Vendor Transaction ID ('AMRTVSQ') program to retrieve the next VENDOR TRANSACTION ID value

.....Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for update to add the record passing this rental contract (ticket) identifier, rental location, the retrieve VENDOR TRANSACTION ID and a CUSTOMER TRANSACTION ID comprised of the current date in ccyyymmdd format.

.....IF its PROGRAM RETURN CODE is not blank, then error the transaction set with ERROR CODE = '82' ("Request Rejected - Cannot Identify Transaction ID").

.....ELSE, Update each loaded record format's key values data elements with the new CUSTOMER TRANSACTION ID and VENDOR TRANSACTION ID values.

.....ENDIF

.....ENDIF

....Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for inquiry to check its STATUS CODE value.

....IF its PROGRAM RETURN CODE is not blank, then error the transaction set with ERROR CODE = '82' ("Request Rejected - Cannot Identify Transaction ID")

....ELSE, its PROGRAM RETURN CODE is blank, then

.....IF the STATUS CODE is not currently 'B' (Billed), 'A' (Audited), 'P' (Paid), 'Q' (Invoice Rejected by Customer), then Execute the ARMS Inquire/Update Rental Cross-Reference Record (AM1010V1) program for inquiry to update its STATUS CODE value to 'B' (Billed).

.....ENDIF

.....ENDIF

....Write the record formats read to the ARMS Distributed Generated Transaction Sets to be sent to the centralized ARMS system for routing file (ANDIST)

....Send the input data queue entry as an output data queue entry to data queue (DQANDST) for subsequent transmission to the centralized ARMS host platform/machine.

.. ..ENDIF

#### @Notes:

This program is submitted for execution by the ARMS Startup Jobs program (CLL810) with no entry parameters to execute only on the ECARS host system platform (NOT "CENTRAL" nor "RARMS"). Currently, the only (non-shutdown) transaction set group type code processed in this program is the proprietary EDI Invoice Transaction Set Invoice ("IN") group types. These invoice transaction sets were generated within a non-ARMS stewarded program, Generate ECARS Daily (Nightly) Billing batch (RB0010) program.

This is currently not performed on the Claims Connection host platform ("CENTRAL") because the Claims Connection process for Closing a Third-Party Rental Ticket sends a data queue entry to the ARMS Dispatch Rental Systems Request (AM0060V1) program to dispatch a data queue entry to the ARMS Generate Proprietary EDI Electronic Invoice Detail Transaction Set (CC00INV1) program, stewarded by ARMS.

IF any error is found, then do not process this transaction any further and execute the ARMS Handle Internal Error (AM0097V1) program, passing the input parameters of this data queue entry's key data elements and the currently loaded

### ARMS Process Report

ERROR CODE value. This will notify ARMS On-Call of the errored transaction set's data queue entry value and the applicable ERROR CODE. Do NOT output the transaction set's record formats to the ANDIST file nor send its data queue entry to DQANDST.

The ARMS system will now enable electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer) and AAA of California. Therefore, ARMS will receive a data queue entry to generate an electronic invoice (and create an associated ARMS Rental Transaction Cross-Reference (AMXREF) file record) when the trading partner never sent to ERAC (via ARMS) any initial rental authorization.

@Files: (CRUD)

ANDQINFO (-R--) ARMS Data Queue Information File  
 AMINTR (-R--) ARMS Unedited Invoice Transaction Sets File  
 ANDIST (CR--) ARMS Distributed Generated Transaction Sets to be sent to the centralized ARMS system for routing.

@Embedded Data/Constants:

'0123456789'  
 '0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ '  
 'AM0062V1'  
 'AM0062V1 '

@Improvement Opportunities:

- 1.) Convert this OPM RPG program to an ILE RPG program.
- 2.) Delete the 'AM0062V1' program name constants and specify/use the Program Status Data Structure's PROGRAM ID data element.
- 3.) Delete the constants for valid numbers and use the TESTN operation for testing numerics.
- 4.) Add the submission of this program for execution on the Claims Connection host platform ("CENTRAL") because the mandatory field data values loaded from its rental system database must also be performed. The transaction set generation and editing processes should only be different in the fact that they access different rental systems databases.
- 5.) Replace the execution of the ARMS Handle Internal Error ('AM0097V1') program with the execution of the newer ARMS Handle Internal/External Error ('AM0098') program.

#### Process

Hierarchical numeric ID: 1.1.2.4.3.4  
 Coded name:  
 Name: DTQ Input to AM0062 (DQAM62V1)  
 Comment: @Definition: DQAM62V1 is a data queue used to provide input for PGM AM0062V1 which processes invoices read from file AMINTR to be sent to an EDI Trading Partner.

#### Process

ARMS Process Report
---------------------

Hierarchical numeric ID: 1.1.2.4.5

Coded name:

Name: BAT Generate Invoice - ECARS

Comment: @Definition: The batch process of generating invoice formats required for electronic billing in response to ECARS closing a rental. This process is triggered by TIME.

@Purpose: To be able to generate invoices for closed rentals in an automated fashion.

#### Process

Hierarchical numeric ID: 1.1.2.4.5.1

Coded name: RB0010

Name: PGM Create ECARS Invoice Detail (RB0010)

Comment: @Purpose:

To generate the initial billing invoices for closed rental contracts/tickets that have billing customers assigned that have a balance due that were closed on the previous business day.

Will also generate duplicate billing invoices for any closed rental contracts/tickets that have billing customers assigned that have a balance due that were selected for rebilling.

#### @Operational Method:

- IF on "DEV" development platform, then retrieve the associated host platform Machine Name for the rental contract/ticket's Group and Branch ID.

ELSE retrieve the current application host platform's Machine Name where this program is executing.

- Set any special machine related attributes.

- For each customer's closed rental tickets selected for generation of the billing invoices in the job file, RACBILLJ (ECR: Facsimile Billing Master File, sorted by Customer), perform the following steps:

1) Retrieve the closed rental contract/ticket information from the closed closed rental contract/ticket and its supplementary files:

2) Retrieve the ARMS profile and attempt to retrieve an associated ARMS Rental Transaction Cross-Reference information by executing the Retrieve ARMS Data (AM2090V1) program.

3) Check if the Bill-To Customer attributes returned from AM2090V1 indicates that the trading partner has the capability to receive electronic billing invoices or should be sent paper billing invoices. (Paper bills are default.)

4) IF electronic billing for ARMS,

a) Generate formats APPD01, IEBH01, ELCD01, RNTD01, REND01.

b) IF the profile dictates that the RATD01 format needs to be generated, generate it with 3 different information contents - units and rates, units only and units and rates without relating to each other.

c) Write the data queue entry to DQAM62V1 to send of the invoice transaction.

d) Delete the original bill from RACBILL.

e) Maintain electronic bills generation counts.

ELSE electronic billing for non-ARMS customer.

a) Generate formats APPD01, IEBH01, ELCD01, RNTD01, and REND01.

b) For invoicing only customers also generate the AUTD01, and ADJD01 formats.

### ARMS Process Report

c) Write the data queue entry to DQAM62V1 to send the invoice transaction.

d) Delete the original bill from RACBILL.

e) Maintain electronic bills generation counts.

ELSE paper bills

a) Print the bill to appropriate medium (tape file, physical file)  
(Tape is turned over to a third party firm for sorting by ZIP/Postal Code, grouping and printing, envelope stuffing, postage and mailing.)

b) Delete the original bill from RACBILL if bill was printed.

c) Maintain paper bill generation counts

5) If unsupported version of billing type (type greater than 4), print on exception report and update the bill in RACBILL to a daily frequency so that the bill may be generated the day the error has been fixed.

6) IF ticket is not found in the closed ticket files, print on exception report and delete the bill from RACBILL.

7) At customer total time, maintain number of envelopes needed and reset paper billing counter.

- Repeat the above steps for the next customer until there are no more customers to process.

@Notes:

The various ticket types are : I=Insurance B=Bodyshop D=Dealer R=Regular  
O=Other.

The various billing types are : 1=Tot Less 2=Dy Tax 3=Dy NoTax 4=Other  
5=Rebill.

The ARMS system now enables electronic invoices, payments and remittances (processed here) to be sent/received for non-ARMS Rental Management Trading Partners that are Direct Billing/Remittance Trading Partners, such as Tennessee Valley Authority (an Enterprise Fleet Services customer). Therefore, at the initial billing invoice generation occurrence, no ARMS Rental Transaction Cross-Reference file record will exist at that time.

@Files: (CRUD)

RACBILLJ	( R )
RABMSG	( R )
FEDTAXID	( R )
QRACCLSQ	( R )
RACSLD	( R )
OFFDRB	( R )
CUSTMAST	( R )
ECSURCHG	( R )
RACBRIST	( R )
RACBILL	( RU )
RACINS	( R )
BCPECG	( R )
AMXCLS	( R )
OFGBLN	( R )
RACILN	( R )
RACDJD	( R )
RACBILLT	( C )
AMINTR	( CRU )
RAGBBT	( R )

# ARMS Process Report

## Process

Hierarchical numeric ID: 1.1.2.4.5.2

Coded name: CHKZP

Name: PGM Format ZIP (CHKZP)

Comment: @Purpose: To place the passed Zip Code into proper country format for printing or display.

### @Operational Method:

This program expects the following parameters: (EDS CHKZPI)

ZPZIPS - SHIPPING Zip Code

ZPEDTS - Edit Code Type

If not provided, format of edited zip code is determined by state.

If provided, determines the format desired for the passed zip code and ignores state code. Valid action codes are:

- |    |                        |         |
|----|------------------------|---------|
| 1: | 5 dash 4 (xxxxx-xxxx)  | U.S.    |
| 2: | 5 blank 4 (xxxxx xxxx) | U.S.    |
| 3: | ** NOT USED **         |         |
| 4: | 5 only (xxxxx )        | MEXICO  |
| 5: | 3 blank 3 (xxx xxx )   | CANADA  |
| 6: | 4 blank 3 (xxxx xxx )  | EUROPE  |
| 9: | 9 blank (xxxxxxxxxx )  | GENERIC |

ZPSTS - State. If passed, and edit code is blank, is used to retrieve proper edit format from state code file.

If state not found, edit format defaults to #9.

ZPNUMS - Edit Code (Blank, "A", "B", "N" or "Z")

Blank = A thru Z & 0 thru 9

A = A thru Z

B = A thru Z, 0 thru 9 & blank

N = 0 thru 9

Z = Anything

ZPZIPM - MAILING Zip Code

ZPEDTM - Edit Code Type

If not provided, format of edited zip code is determined by state.

If provided, determines the format desired for the passed zip code and ignores state code. Valid action codes are:

- |    |                        |         |
|----|------------------------|---------|
| 1: | 5 dash 4 (xxxxx-xxxx)  | U.S.    |
| 2: | 5 blank 4 (xxxxx xxxx) | U.S.    |
| 3: | ** NOT USED **         |         |
| 4: | 5 only (xxxxx )        | MEXICO  |
| 5: | 3 blank 3 (xxx xxx )   | CANADA  |
| 6: | 4 blank 3 (xxxx xxx )  | EUROPE  |
| 9: | 9 blank (xxxxxxxxxx )  | GENERIC |

ZPSTM - State. If passed, and edit code is blank, is used to retrieve proper edit format from state code file.

If state not found, edit format defaults to #9.

ZPNUMM - Edit Code (Blank, "A", "B", "N" or "Z")

Blank = A thru Z & 0 thru 9

A = A thru Z

B = A thru Z, 0 thru 9 & blank

N = 0 thru 9

Z = Anything

### ARMS Process Report

The program returns the following parameters: (EDS CHKZPO)

ZOZIPS - SHIPPING Zip code, formatted for print. 10 pos.  
 ZOCNTS - Country code. Retrieved from State file if not passed.  
 ZONAMS - Country Name. Retrieved from State file.  
 ZOMSGS - Error Message. If field is greater than blank, indicates an error.  
 ZOZIPM - SHIPPING Zip code, formatted for print. 10 pos.  
 ZOCNTM - Country code. Retrieved from State file if not passed.  
 ZONAMM - Country Name. Retrieved from State file.  
 ZOMSGM - Error Message. If field is greater than blank, indicates an error.

-----  
 @Notes: Program moves input ZIP to output ZIP before performing edits. This serves as a default in case that an error occurs, program will always return a value. Calling program should but is not forced to look at the error message.

#### Process

Hierarchical numeric ID: 1.1.2.4.5.3  
 Coded name: VTS012A  
 Name: PGM Retrieve Vehicle Licensing Fee Charge Detail (VTS012A)  
 Comment: @Purpose:

To retrieve the Vehicle License Fee Days, Daily Unit Charge, and Total Charge for all input units for a group, branch, and ticket number.

#### @Operational Method:

The program expects the following data elements as the parameters (EDS VT024X1) with group, branch, ticket and the four unit numbers filled in:

Processing Mode  
 Ticket  
 Calendar Rental Flag  
 Ticket End Date  
 Renting Group  
 Renting Branch  
 Init Total Days  
 Unit Total Days  
 Unit 1  
 Unit 2  
 Unit 3  
 Unit 4  
 Date Out 1  
 Date Out 2  
 Date Out 3  
 Date Out 4  
 Date In 1  
 Date In 2  
 Date In 3  
 Date In 4  
 Init VLF Days Unit 1  
 Init VLF Days Unit 2  
 Init VLF Days Unit 3  
 Init VLF Days Unit 4  
 New VLF Days Unit 1  
 New VLF Days Unit 2  
 New VLF Days Unit 3



**ARMS Process Report**

New VLF Days Unit 4  
Segment 1 Days  
Segment 2 Days  
Segment 3 Days  
Segment 4 Days  
Segment 1 Weeks  
Segment 2 Weeks  
Segment 3 Weeks  
Segment 4 Weeks  
Segment 1 Months  
Segment 2 Months  
Segment 3 Months  
Segment 4 Months  
Daily VLF Unit 1  
Daily VLF Unit 2  
Daily VLF Unit 3  
Daily VLF Unit 4  
Total VLF Unit 1  
Total VLF Unit 2  
Total VLF Unit 3  
Total VLF Unit 4  
Total Rental VLF  
CA Branch  
Employee Number  
Start Charges 1 Time  
Ticket end time  
VLF Bill To Flag  
CA License Renewal No-Unit 1  
CA License Renewal No-Unit 2  
CA License Renewal No-Unit 3  
CA License Renewal No-Unit 4  
Bill To Less VLF  
Bill To Plus VLF  
Return Code 1  
Return Code 2  
Return Code 3  
Return Code 4

**Process**

Hierarchical numeric ID: 1.1.2.4.5.6

Coded name: CCRAVN

Name: PGM Retrieve VIN (CCRAVN)

Comment: @Purpose: To return the Vehicle Identification Number (VIN) for a unit.

**@Operational Method:**

The program expects the following input parameters: unit number, ticket type ('D'ealer) and Option ('1'= Open a Ticket). VIN is returned along with the print flag. The parameter list is as follows:

UNIT NUMBER  
TICKET TYPE  
VIN#  
PRINT FLAG  
TICKET OPTION

**Process**

**ARMS Process Report**

Hierarchical numeric ID: 1.1.2.4.5.7

Coded name:

Name: RPT Exceptions of ECARS Rental Contracts Not Invoiced

Comment: @Purpose: To create a report of all the ECARS closed tickets that were not invoiced during this invocation of the invoice generator, RB0010, because of an error so that it can be investigated and the error fixed before the ticket is resubmitted for invoicing.

@Report Distribution: Not defined.

Process

Hierarchical numeric ID: 1.1.2.4.5.8

Coded name: CLJ340

Name: PGM Driver Program for Invoice Generation (CLJ340)

Comment: @Purpose: Prepare and bill closed / re-billed tickets for all trading partner based on their frequency.

@Operational methods:

- Collect requests for any ticket to be re-billed.
- Total the number of records for a given control group
- Update the detail records with the total for the control group
- Generate the invoices
- Generate listing of invoices not created.

Process

Hierarchical numeric ID: 1.1.2.4.5.9

Coded name: RB0030

Name: PGM Create Total Record for Control Group (RB0030)

Comment: @Purpose: Create a total record for each control group, to be used later in the process for sorting and processing.

@Operational Method:

- Read every detail record, incrementing a number per control group.
- Write a record to work file with the total for control group.

@Notes: This pertains to invoices that are faxed.

Process

Hierarchical numeric ID: 1.1.2.4.5.10

Coded name: RB0031

Name: PGM Update Detail Records with Control Group Total (RB0031)

Comment: @Purpose: Update detail records with control group total.

@Operational Method:

- Read every detail record, updating it with the control group total

@Notes: This pertains to invoices that are faxed.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.2.4.5.11  
Coded name: RB0037  
Name: PGM Calculate the actual number of Facsimilie (RB0037)  
Comment: @Purpose: Update the individual detail records with the number total number control records for a givien group.

**@Opertaional Method:**

- Read the file counting the number of control records for a given group
- When the control group changes, update each detail record with that number.

**@Notes:**

- This number is used for determining if a cover page is to be generated with the invoices.

**Process**

Hierarchical numeric ID: 1.1.2.4.5.12  
Coded name: F20114  
Name: PGM Generate Facsimile Spool File for Control Group (F20114)  
Comment: @Purpose: Distribute the spool file based on language code

**@Operational Method:**

- Read to work file group all records for a customer together.
- When the total page count is less than 6 no cover page is generated.
- When the total page count is 6 - 9, generate cover page - Portrait.
- When the total page count is greater than 10, generate cover page - landscape.
- Generate report.

**Process**

Hierarchical numeric ID: 1.1.2.5  
Coded name:  
Name: BT Close ARMS Authorized Ticket  
Comment: @Definition: The set of online activities that are used to calculate the billing totals for an ARMS Authorized Ticket that has closed.

@Purpose: Process that is executed to calculate the Trading Partner amount due.

**Process**

Hierarchical numeric ID: 1.1.2.5.2  
Coded name:  
Name: ONL Close ARMS Authorized ECARS Ticket  
Comment: @Definition: The process that close goes through to calculate the billing charges for an ARMS ticket.

**Process**

<b>ARMS Process Report</b>
----------------------------

Hierarchical numeric ID: 1.1.2.5.2.1

Coded name: ECARMSV1

Name: PGM Calculate EC Billing Charges for ARMS Tickets (ECARMSV1)

Comment: @Purpose: To calculate the amount to be billed to the renter and the amount to be billed to the insurance company at closing time for an ARMS authorized ticket.

**@Operational Method:**

This program when called will receive the ticket information as parameters.

For the passed ticket, perform the following steps:

- 1) Calculate the number of days to bill and total days authorized.
- 2) Get the group branch tax information to determine what surcharges apply and if the surcharges are taxable.
- 3) Load the approved surcharges from the file ECSURCHG, Surcharges Approved by Insurance Company File, into arrays based on the surcharge type, i.e., daily, lump and percent. Add the daily rate augmentor surcharges ('02'=city/state and '20'= underage) to the calculated authorized daily rate. Similarly, add the lump surcharge with code '11'=Miscellaneous to the calculated authorized lump total field. These special surcharges are not loaded to the surcharge arrays.
- 4) Check for any government surcharge, airport charge, additional charges (baby seat, additional driver charge etc) per rental charged on the ticket against the lump array to see if the insurance company has authorized that surcharge. If the amount in the ticket information matches the amount in the array, it means that this surcharge amount was authorized by the insurance company as the arrays were loaded from the approved surcharges file. If authorized, add this to the calculated lump total or the taxable lump total field if the charge is taxable for this ticket.
- 5) Check for any government surcharge, airport charge, additional charges (baby seat, additional driver charge etc), damage waiver for each unit, personal accident insurance for each unit, per day, charged on the ticket against the daily array to see if the insurance company has authorized that surcharge. If the amount in the ticket information matches the amount in the array, it means that this surcharge amount was authorized by the insurance company as the arrays were loaded from the approved surcharges file. If authorized, add this to the calculated daily authorized rate total or the dialy taxable total field if the charge is taxable for this ticket.
- 6) Similarly, check for percentage surcharges in the percent array.
- 7) Apply any discount to the calculated rates and compare against authorized maximum amounts for the ticket. If the calculated charges exceed the maximums, select the maximums as the charges to be billed to the insurance company.
- 7) If the claim type is "claimant", policy maximums are ignored otherwise the lesser of the bill-to amount and the policy maximum is chosen.
- 8) Retrun with the amount to be billed to the insurance company and the amount to be billed to the renter.

@Notes: Some ARMS Trading Partner Companies does not send us the amounts for the surcharges. They simply send us flags indicating which surcharges they will pay for ,e.g., PAI, DW, SLP etc. Any charge that the branch charges for the surcharge flag sent by them will be considered authorized subject to the maximums.

**@Files:**

GPBRTX	( R )
ECSURCHG	( R )
VT010P00	( R )
OFFDRB	( R )

ARMS Process Report
---------------------

## Process

Hierarchical numeric ID: 1.1.2.5.3

Coded name:

Name: ONL Close Claims Connection Ticket

Comment: @Definition: The process that close goes through to calculate the billing charges for an ARMS ticket.

## Process

Hierarchical numeric ID: 1.1.2.5.3.1

Coded name: CCARMSV1

Name: PGM Calculate CC Billing Charges for ARMS Tickets (CCARMSV1)

Comment: @Purpose: To calculate the amount to be billed to the renter and the amount to be billed to the insurance company at closing time for an ARMS authorized ticket.

## @Operational Method:

- 1) Retrieve the ticket data.
- 2) Determine how many and which rental locations were used.
- 3) For each location perform the following steps:
  - \* Calculate number of billing days for this location.
  - \* Determine Bill-To days and the remaining authorized days
  - \* Determine Bill-To Daily Rate
  - \* Calculate total surcharge amount for this ticket/location
  - \* Calculate Daily Maximum
  - \* Select the lesser of the daily rate or the daily maximum
  - \* Multiply bill-to daily rate times the number of bill-to days to determine extended bill-to rate.
  - \* Add taxable authorized lump totals
  - \* Calculate tax percentage. Multiply extended bill-to rate time tax percent to determine bill-to taxable amount
  - \* Calculate markup per day amount, extend it and add to the bill-to amount
  - \* Add in percentage surcharges to total bill-to.
  - \* Percentage surcharges are non-taxable and are applied only to the base amount
  - \* Add approved non-taxable lump, daily, and percentage surcharges to the total bill-to ticket amount.
  - \* If loss type is NOT a claimant, select the lesser of the authorized amount and the calculated bill-to amount.
  - \* Calculate the bill-to percentage and the amount due from the trading partner.
  - \* If loss type is NOT a claimant, select the lesser of the amount due from the trading partner or the policy maximum.

## @Files:

CCCLSC	( R )
CCMASTER	( R )
CCSURDTL	( R )

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.3

Coded name:

Name: AA Office Information Synchronization Between Enterprise and Trading Partner

Comment: @Definition: The application area responsible for keeping the Enterprise office information database on the insurance companies host computer in synch with the database on Enterprise computer with regard to any new additions, changes or deletions to the office or the predefined rental rates information.

**Process**

Hierarchical numeric ID: 1.1.3.1

Coded name:

Name: BT Synchronize Office Information for specified ARMS Trading Partner Companies

Comment: @Definition: The set of automated activities that are used to create and transmit Rental Location office information to a specified ARMS Trading Partner Company's host computer.

@Purpose: Communicate Enterprise existing rental location office information (telephone and/or rates) to the trading partner.

**Process**

Hierarchical numeric ID: 1.1.3.1.5

Coded name:

Name: BAT Generate Office Maintenance Transactions for specified ARMS Trading Partner Companies

Comment: @Definition: The batch process that generates required office and predefined rental rates maintenance transactions to be sent to specified ARMS Trading Partner Companies host computer. This batch is triggered by TIME and scheduled in the automated job scheduler.

@Purpose: To allow for timely updated information of office and rate info to specified ARMS Trading Partner Companies.

@Notes: Rental Location Office changes are made by the Rental Application Systems during the day. These changes are process when this processes runs.

**Process**

Hierarchical numeric ID: 1.1.3.1.5.1

Coded name: CLL900

Name: PGM Distribute Office and Rate Updates (CLL900)

Comment: @Purpose:

To collect office and predefined rental rates information changes and generate maintenance transactions to synchronize this information on the insurance company's host computer. Also, to consolidate these transactions from the distributed systems to the centralized ARMS system.

@Operational Method:

1) CALL CLL042 to generate rate maintenance transactions in file AMLOCTR, Location/Rate Change Transactions.

**ARMS Process Report**

2) IF the returned run status is 'Y', CALL AM0041V1 to generate the ARMS proprietary formats and send them to the centralized system.  
ELSE end the program.

**Process**

Hierarchical numeric ID: 1.1.3.1.5.2  
Coded name: CLL042  
Name: PGM Gather Rate Updates (CLL042)  
Comment: @Purpose:

To determine if there are any changes to the predefined rental rates in file NRXRATES (national reservation exception rates) for insurance type customers and to generate synchronizing transactions for maintaining the rates on specified ARMS Trading Partner Company's host computer.

**@Operational Method:**

- Retrieve job run date and add one day to get tomorrow's date. Subtract one day to get yesterday's date.

1) Retrieve the last job completed date from data area DAL900STS and compare this date to yesterday's date.

a) IF this is not a rerun and the dates are not equal, notify ARMS On-Call (backup and primary). Set the run status flag to 'N' and return, so that the calling program, CLL900, does no further processing.

b) IF this is a rerun or the dates are equal (i.e. the job was completed yesterday), continue.

2) Open a query file selecting those rates from NRXRATES that become ineffective today and those that become effective tomorrow. Then CALL AM0042 to create rate maintenance transactions in AMLOCTRN for these selected records.

**Process**

Hierarchical numeric ID: 1.1.3.1.5.3  
Coded name: AM0042  
Name: PGM Create Rate Update Transactions (AM0042)  
Comment: @Purpose:

To create predefined rental rate maintenance transactions for a location. Each location may have rates specified for each type of car that the location hosts with effective dates.

**@Operational Method:**

1) Read NRXRATES (Query file over NRXRATES, national reservations exception rates opened by CLL042) in a loop until end of file. The selection criteria for the query are: {Pre-arranged discount type = 'I' (Insurance)} AND {(Start Date = Job Date + 1) OR (Stop Date = Job Date)}. This will select any rates for a location that are ending today and/or any rates that are effective tomorrow.

2) In order to determine which type of maintenance (add, change or delete) to perform; this program holds the previous records key while it sequentially reads the query file (Key: Group + Branch + Car Type + Start Date). Remember we are dealing with only those records that are selected by the query.

ARMS Process Report
---------------------

a) IF the current record's Group+Branch+Car Type matches the previous record's, then this signifies a 'C'hange. Create rate change transaction in AMLOCTRN for this location and car type.

b) IF the current record's Group+Branch+Car Type does not match the previous record's, then :

i) IF the start date is tomorrow, this is an 'A'dd. Create rate add transaction in AMLOCTRN for this location and car type.

ii) IF the stop date is today, this is a 'D'elete. At this point, this condition HAS to be true. Create rate delete transaction in AMLOCTRN for this location and car type.

@Files:

NRXRATES (R\_)  
NRCTY (R\_)  
AMLOCTRN (C\_)

### Process

Hierarchical numeric ID: 1.1.3.1.5.4

Coded name: AM0041V1

Name: PGM Format and Distribute Office and Rate Updates (AM0041V1)

Comment: @Purpose:

To read the office and rate change transactions and generate appropriate ARMS proprietary formats for the synchronizing maintenance transactions that need to be sent to the insurance customer.

@Operational Method:

1) Read the ARMS Profile Application Specific Data (ARMSPR1) file starting with the first record and for each profile that specifies that the insurance company receives office and rate updates, perform the following steps.

2) Read the AMLOCTRN file and for each location in this file, check to make sure that the location is set up to receive ARMS reservations. IF the location is not set up, skip this record and read the next record ELSE continue.

3) Get the effective predefined rental rates for this location from the NRXRATES file and generate the office and rate maintenance transaction sets (format OFFD01) for this profile and write to file ANDIST for transmission to the centralized system. A data queue entry to the data queue DQANDST will trigger the transmission.

4) Get next record from AMLOCTRN and repeat steps 2,3,4 until all records in AMLOCTRN are read.

5) Finally, delete all records from file AMLOCTRN.

@Files:

ARMSPR1 (R\_)  
AMLOCTRN (CR\_D)  
NRXRATES (R\_)  
DROFF (R\_)  
NRCTY (R\_)  
ANDIST (C\_)

### Process



**ARMS Process Report**

Hierarchical numeric ID: 1.1.3.2

Coded name:

Name: BT Synchronize Office Information for X12 Customer

Comment: @Definition: The set of automated activities that are used in the creation and transmission of Rental Location Office information to be sent to X.12 customers .

@Purpose: Communicate Enterprise existing rental location office information to the trading partner.

**Process**

Hierarchical numeric ID: 1.1.3.2.6

Coded name:

Name: BAT Generate Office Location / Rates Maintenance Transaction Set 832 for X12

Comment: @Definition: The batch process of generating required office maintenance transactions (which includes office information and rates information) to be sent to X.12 customers. This batch is triggered by TIME and scheduled in ROBOT.

@Purpose: To allow for timely updated information of office and rate info to X-12 customers.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.1

Coded name: AML832A001

Name: MOD Convert Journal Entries to Transactions (AML832A001)

Comment: @Purpose:

To convert office directory information journaled changes to a change transactions for further processing.

@Operational Method:

- Read the job file, AML832JA (created via the DSPJRN command in the calling procedure) sequentially.

- Based on the before and after images of the office directory records in the journal entries, construct the change transactions with the appropriate business case into file, AM830P (AM Office change transactions). This will handle two business cases :

- 1 = Add/Update Rental Car Company Branch Number
- 6 = Delete Branch Number

@Notes:

1. Only U.S. Daily Rental locations will be processed.
2. If an error occurs, generate an error/page, then return and 'Exit Code' of '1' so the caller can detect that an error occurred.

Note also that this program module reads from AML832JA which is a file created by the DSPJRN command.

@Files:

AM830P (C\_UD)  
MACHID (\_R\_)

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.3.2.6.2  
Coded name: AML832A002  
Name: MOD Determine Company & Invoke Send Office Updates (AML832A002)  
Comment: @Purpose:

To determine which companies have requested for the office and pre-defined rental rate change transactions to be sent to them and to send the change transactions to the company if requested.

**@Operational Method:**

1) Read the profile data to determine which company has requested to receive the X.12 synchronizing office and rate maintenance transactions.

2) For each company that has requested office updates, perform the following (procedure ProcCstOfcUpdts):

a) Build the beginning transmission envelope using procedures BldArmsKey, BldTSMS01, BldGRPS01 and BldSETS01 in that order into AMSEND.

b) Invoke the GetCarClasses procedure with appropriate parameters to get the Enterprise and X12 car classes for this customer from file AMCLSTBL.

c) For all records in file AM830P (Office change transactions), invoke the procedure FmtSndOfcUpdts with appropriate parameters to write office update records to AMSEND. If a new office location is being added and the GetCarClasses procedure returned car classes, invoke the FmtSndRateUpdts procedure with appropriate parameters to write Rate Format records to AMSEND for the customer.

d) Build the ending transmission envelope using procedures BldSETB01, BldGRPE01 and BldTSME01 in that order into AMSEND.

e) Send a Data Queue to DQ832S to initiate the sending of this transmission.

**@Notes:**

If an error occurs, generate an error/page, then return an 'Exit Code' of '1' so the caller can detect that an error occurred. If an 'Exit Code' of '1' is returned from a called procedure, return the 'Exit Code' to the caller immediately.

**@Files:**

AM830P (-R--)  
AM831P (-RU-)  
ARMSPR1 (-R--)  
ARMSPR3 (-R--)  
AMSEND (C---)

**Process**

Hierarchical numeric ID: 1.1.3.2.6.3  
Coded name: AML832A003  
Name: MOD Format and Send Office Updates (AML832A003)  
Comment: @Purpose:

### ARMS Process Report

To format the information passed from the transaction file record to prepare it for sending.

@Operational Method:

Procedure FmtSndOfcUpdts

- 1) Use procedure BldArmsKey to build the key for AMSEND.
- 2) Use procedure BldLocFmt (Build location format) to build the office maintenance transaction proprietary format X832F2 and write to AMSEND.
- 3) IF the location format is a business case 6 (delete branch), additional formats need to be generated to handle the service area updates as a result of the delete.
  - a) Retrieve all the service areas served by this branch (rental location) by reading the service area/location file AM831P01 in a loop using the rental location as the key.
  - b) For each of these service areas, call the locator procedure, AMRtnUSLoc, passing the service area to get the new branch for this service area. Update service area file, AM831P01 with the new location returned from AMRtnUSLoc unless a return code of '06' is received or the location returned is blank. In these cases, generate an error/page and skip to the next service area.
  - c) Use procedure BldSvcFmt to create the service area update format X832F1 and write to AMSEND.

@Files:

AMSEND (C---)  
AM831P (-RU-)

@Notes:

1. The service area formatting is shared with job AML833A and as such must be considered for a service program.
2. In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred. If an 'Exit Code' of '1' is returned from a called procedure, return the 'Exit Code' to the caller immediately. There may be alternative handling of some errors noted in the operational method.

Process

Hierarchical numeric ID: 1.1.3.2.6.4  
Coded name: AML832A004  
Name: MOD Maintain data area AM832A (AML832A004)  
Comment: @Purpose:

To maintain data area AM832A with the starting (from) date and time to be used for filtering the DSPJRN command in program AML832A on the next scheduled run. This is the last task performed after all processing for the job has been completed.

@Operational Method:

Procedure SetNxtFrmDtTm.

Retrieve and lock data area AM832A. It will contain the starting (from) date and time used in the DSPJRN command for the current job.

Add 1 second to the date and time passed to this procedure, which

**ARMS Process Report**

will contain the ending (to) date and time used to filter the DSPJRN command for the current job.

Update data area AM832A with the new value and unlock.

**@Notes:**

In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.5  
Coded name: AML832A005  
Name: MOD Format and Send Rates/Car Classes (AML832A005)  
Comment: @Purpose:

To format the rates for retrieved car classes for any new Enterprise location being sent to the specified insurance company.

**@Operational Method:**

Procedure FmtSndRateUpdts.

1) Use procedure GetClassRates to get the rates for the specified customer for the specified location.

2) Use procedure SndRateFmts to write rate format X832F3 records to AMSEND for each X12 Class and Rate. Handle return codes as follows:

ALLRATES '00' Rates for all classes were found and sent for loc.  
INVALPARM '01' Generate error/page and process next location.  
NORATES '02' Generate error/page and process next location.  
SOMERATES '03' Rates for some of the classes were found and sent.  
LOCKEDRECORD '99' Generate error/page and process next location.

**@Files:**

AMCLSTBL (-R--)  
AMSEND (C--)

**@Notes:**

1) In general, if an error occurs generate an error/page; then return an 'Exit Code' of '1' so the caller can detect that the error occurred. There may be alternative handling of some errors noted in the operational method.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.6  
Coded name: AML832A006  
Name: MOD Build ARMS Start/End for Office Updates (AML832A006)  
Comment: @Purpose:

**ARMS Process Report**

To build the key and data fields for the transmission, group, and set start and end envelope for office and rate updates.

**@Operational Method:**

The following procedures are exported for use by other procedures. When building the office and rate update transactions, these procedures provide the ARMS proprietary transmission envelope formats.

The file AMSEND consists of two parts, the key and the proprietary format. These procedures will return the key and envelope proprietary formats in the AMSEND data structure, AMSENDDS.

**1) Procedure BldArmsKey:**

- a) Clear the ARMSKEY and AMSEND data structures.
- b) Populate the ARMSKEY (Data Structure) fields and move the ARMSKEY data structure to the ASNKEY field of the AMSEND data structure
- c) Return

**2) Transmission Start procedure BldTSMS01:**

- a) Clear the TSMS01 data structure.
- b) Populate the TSMS01V1 (Data Structure) fields and move the TSMS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**3) Group start procedure BldGRPS01:**

- a) Clear the GRPS01 data structure
- b) Populate the GRPS01V1 fields and move the GRPS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**4) Set start procedure BldSETS01:**

- a) Clear the SETS01 data structure
- b) Populate the SETS01V1 fields and move the SETS01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**5) Transmission end procedure BldTSME01:**

- a) Clear the TSME01 data structure
- b) Populate the TSME01V1 fields and move the TSME01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**6) Group end procedure BldGRPE01:**

- a) Clear the GRPE01 data structure
- b) Populate the GRPE01V1 fields and move the GRPE01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**7) Set end procedure BldSETE01:**

- a) Clear the SETE01 data structure
- b) Populate the SETE01V1 fields and move the SETE01V1 data structure to the ASNDTA field of the AMSEND data structure
- c) Return

**@Notes :**

- 1. This module has been bound by copy into two different ILE programs AML832A and AML833A and such must be converted into a service program at some point.

**ARMS Process Report**

2. If an error occurs, generate an error/page, then return an 'Exit Code' of '1' so the caller can detect that an error occurred.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.7  
Coded name: AML833A001  
Name: MOD Select Unique Service Areas (AML833A001)  
Comment: @Purpose:

To select each unique ARMS Trading Partner Company service area (area code + exchange) from all the phone numbers in the US, associate a location, and record the combination.

**@Operational Method:**

1) The first parameter to this module specifies the service area to start processing from. If it is blank, process all service areas.

2) Since the service areas consist of the phone number, area code, and prefix; the maximum service area is '999999'. For each sequential possible service area from 1 through 999999, try positioning file AT001P01 to the record for that service area. If successful, send a data queue containing that service area to DQ834Q.

3) Monitor data queue DQ834Q while processing. Any time the data queue accumulates more than 1000 entries, delay the job for 45 seconds repeatedly until there are 1000 or less entries. If 30 delays are executed before the count falls to 1000 or less, return a value of 'Y' (Abnormal end) to AML833A so it will generate an error/page.

4) After service area '999999' is processed, return to AML833A.

**@Files:**

AT001P01 (-R--)

**Process**

Hierarchical numeric ID: 1.1.3.2.6.9  
Coded name: AML833A003  
Name: MOD Determine Company and Invoke Send SA Updates (AML833A003)  
Comment: @Purpose:

To determine which companies have requested to receive the service area updates and then invoke the send routine for each such company.

**@Operational Method:**

1) Determine which company has requested to receive office and rate changes and for each such company, invoke procedure SndSvaUpd to send the updates.

**@Notes:**

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

**@Files:**

<b>ARMS Process Report</b>
----------------------------

ARMSPR1 (-R--)  
ARMSPR3 (-R--)

**Process**

Hierarchical numeric ID: 1.1.3.2.6.10  
Coded name: AML833A004  
Name: MOD Format and Send SA Updates (AML833A004)  
Comment: @Purpose:

To format and send the service area change transactions to the specified company in the input parameters.

**@Operational Method:**

Procedure SndSvaUpd using specified company.

- 1) Use procedure BldArmsKey to build the key portion in file AMSEND.
- 2) Use transmission envelope procedures BldTSMS01, BldGRPS01 and BldSETS01 to build the beginning envelope.
- 3) For each service area in file, AM833P (AM Changed Service Areas) build proprietary format X832F1 with the appropriate business case into file AMSEND.
- 4) Build the ending envelope using procedures BldGRPE01, BldSETE01 and BldTSME01.
- 5) Send a data queue entry with the ARMSKEY to data queue DQ832S. This data queue entry will be picked up by the send interface program for actual mapping and sending.
- 6) Return to caller.

**@Notes:**

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

**@Files:**

AM833P (-R--)  
AMSEND (C---)

**Process**

Hierarchical numeric ID: 1.1.3.2.6.11  
Coded name: AML833A005  
Name: MOD Return a U.S. Location (AML833A005)  
Comment: @Purpose:

To return a U.S. location for a passed service area (Area Code and Exchange)

**@Operational Method:**

- 1) Initialize the data structure that is passed to the locator program, RAS013A.

**ARMS Process Report**

- 2) Execute appropriate overrides to make the locator retrieve Enterprise branches only.
  - 3) Use the locator program to determine the best case rental branch that could serve the service area passed to the program.
  - 5) To determine if the returned branch is usable, we check DROFLF1.  
IF prevent ARMS, use forward to location.  
IF No forward to location, keep looking.  
IF country code is 'CA', keep looking.
  - 6) The return value for procedure AMRtnUSLoc should be 12 characters. The first 2 positions are the return code (same codes as RAS013A). The other 10 positions are the location (as returned from RSA013A).
  - 7) If there is an unexpected error, use \*PSSR to page on call staff after dumping the program
  - 8) RETURN.
- @Files:
- |         |        |
|---------|--------|
| DROFLF1 | (-R--) |
| DROFFL7 | (-R--) |

**Process**

Hierarchical numeric ID: 1.1.3.2.6.12  
Coded name: AML833A  
Name: PGM Send Service Area Updates (AML833A)  
Comment: @Purpose:

To control the various procedures to synchronize the service areas (area code + exchange) representing Enterprise locations in the ARMS Trading Partner Company database with the Enterprise office directory database.

**@Operational Method:**

- 1) The first input parameter specifies the number of never ending jobs which this program submits; each of these jobs calls program AML834A (Process Service Areas). These jobs will share the processing of entries sent to data queue AM834Q by this program (module AML833A001) by repeatedly receiving the next available data queue entry and processing it. This method is used to speed up processing.
- 2) If the second parameter is blank, this is not a restart; clear files AM833P (changed service areas) and AM834P (Claims Connection service areas).
- 3) Once the jobs are submitted, call procedure AML833A001 to process service areas. The second parameter to this program specifies the starting service area from where this procedure will start processing in the global phone number file. If it is blank, all service areas will be processed. The procedure will sequentially select all service areas from the starting point and send them to data queue AM834Q for processing by the AM834A jobs.
- 4) Generate an error/page if AML833A001 returns a value of 'Y' (Abnormal end). Otherwise, send a \*DOWN data queue to each of the AML8334A jobs to end them.
- 5) End this program when all the submitted jobs have shutdown.



ARMS Process Report
---------------------

## @Notes:

- 1) This job runs only on the ARMS Centralized system.
- 2) This program clears file AML833DA but the file is not used for anything. It is in the evolution plan to remove it and update the comments in all associated modules.
- 3) If an error occurs or an 'Exit Code' of '1' is returned by a called procedure, generate an error/page and end the program immediately.

## Process

Hierarchical numeric ID: 1.1.3.2.6.13  
 Coded name: AML832A  
 Name: PGM Update Office Directory (AML832A)  
 Comment: @Purpose:

This is a CLLE program module and will serve as the entry point module for the ILE program AML832A which will consist of bound modules that will synchronize the office information between Enterprise and X.12 customers (i.e., any ARMS Trading Partner Company).

## @Operational Method:

- 1) Make sure the machine that this module executes is "RARS". If not RARS, end the program.
- 2) Get the last process date and time for the job from the data area AM832A. This is where the last process ran left off and this run should now get the entries from this date and time to the current date and time.
- 3) Convert the journal entries available in the OMS journal, @JRNLIB/OMSJRN for file OFFDRB to a job file, AML832JA. The job file has been created using command DSPJRN and essentially has the format the same as QADSPJRN (IBM supplied outfile format) except that the last field called JOESD (Entry Specific Data) is of length 480 which is the record length for OFFDRB. The command syntax will be as follows:  

```
.. DSPJRN JRN(@JRNLIB/OMSJRN) FILE((*LIBL/OFFDRB *FIRST)) RCVRNG(*CURCHAIN)
FROMTIME(&FROMDATE &FROMTIME) TOTIME(&TODATE &TOTIME) JRNCDE((R *ALLSLT))
OUTPUT(*OUTFILE) OUTFILEMT(*TYPE1) OUTFILE(QTEMP/AML832JA) ENDTALEN(480)
```

The field JOESD will contain the after image of the entire OFFDRB record. Journal code (JRNCDE) of "R" means only those journal entries that relate to operations at the record level (add, update, delete). The journal receiver range needs to be \*CURCHAIN which means that all the receivers that are ON LINE since the last break in the receiver chain. OMS deletes OMSJRN journal receivers when they are 48 hours old. A new journal receiver is created whenever the current receiver is 200 Megabytes in size and a new receiver is created every day whether the last receiver reaches its threshold or not.
- 4) CALLPRC CnvtJrn1ToTrns (Module AML832A001) to convert the journal entries to change transactions.
- 5) CALLPRC PrcOfcUpdates (Module AML832A002) which will determine all the companies that have requested office information to be sent to them and will then invoke the send module for each company.

**ARMS Process Report**

6) CALLPRC SetNxtFrmDtTm (Module AML832A004) with the converted 'to date' and 'to time' in CYMD and HMS format. This procedure will maintain the data area AM832A for processing dates.

**@Notes:**

If an 'Exit Code' of '1' is returned from a called procedure or an error occurs, generate an error/page and end the program immediately.

This program module creates the file AML832JA as a result of the DSPJRN command.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.15

Coded name:

Name: DTQ Service Areas to Send - Input to Sender (DQ832S)

Comment: @Definition: The data queue that initiates the sending of the X.12 synchronizing office and rate maintenance transactions to the X.12 insurance customer.

**Process**

Hierarchical numeric ID: 1.1.3.2.6.16

Coded name: AML834A

Name: PGM Process Service Areas (AML834A)

Comment: @Purpose:

To create a service areas update file containing service areas that have been changed since the last time these service areas were sent to the insurance customers.

**@Operational Method:**

- 1) Receive the service area from the input data queue, AM834Q.
- 2) Use procedure AMRtnUSLoc to get the Enterprise US location that will service this service area.
- 3) IF the service area exists in file AM831P (all US service areas file)
  - a) if the location on file is the same as the location returned by AMRtnUSLoc, skip this service area else update the file with the correct location.
  - b) add this service area to file, AM833P (changed service areas file)
  - c) if the returned location is Claims Connection, add the service area to file AM834P (Claims connection service areas).
- 4) IF the service area does not exist in file AM831P (all US service areas file)
  - a) add the service area to file AM831P.
  - b) add this service area to file, AM833P (changed service areas file)
  - c) if the returned location is Claims Connection, add the service area to file AM834P (Claims connection service areas).
- 5) When a shutdown entry is received from the data queue, update data area AM832A to reflect that this job is shutting down.

**ARMS Process Report****@Files:**

AM831P (CRU-)  
AM833P (CR--)  
AM834P (C---)

**Process**

Hierarchical numeric ID: 1.1.3.2.6.17  
Coded name:  
Name: DTQ Service Areas to Process (AM834Q)  
Comment: @Definition: This data queue is used to feed job AML834A with unique service areas for all US phone numbers. There can be several AML834A jobs running that feed off of this data queue.

**Process**

Hierarchical numeric ID: 1.1.3.3  
Coded name:  
Name: BT Send Initial or Yearly Car Class Rates and Initail Office Information for X12 Customer  
Comment: @Definition

The initial/yearly process of sending car class rates and office information to insurance customers who recieve this information in X.12 format.

**@Purpose**

Provide insurance customers with Enterprise office information and car class rates initially. Provide them with car class rates both initially and at yearly intervals.

**Process**

Hierarchical numeric ID: 1.1.3.3.1  
Coded name: AML831A  
Name: PGM Send Car Class Rates (Initial/Yearly) and Office Info (Initial) (AML831A)  
Comment: @Purpose

To control the procedure to send initial or yearly rates for car classes and initial office information to insurance customers who receive this information in X.12 format.

**@Operational Method:**

This program only runs on RARMS or DEV. On other computer platforms, it will exit immediately without processing.

Call procedure AML831A001 to retrieve rates and send them for the insurance customer specified.

**Process**

Hierarchical numeric ID: 1.1.3.3.2

**ARMS Process Report**

Coded name: AML831A001

Name: MOD Send Rates (Initial/Yearly) and Office Info (Initial) (AML831A001)

Comment: @Purpose

To send initial or yearly car class rates and initial office locations to the specified insurance customer in X.12 format.

**@Operational Method**

- 1) Retrieve a transmission ID from data area AMTCID.
- 2) Call procedures in module AML832A006 to build the key and write the start formats to file AMSEND.
- 3) Call procedure GetCarClasses (module AML832A005) to retrieve car classes for the customer.
- 4) Call procedure LoadMachArray to retrieve all U.S. computer platforms.
- 5) Read through file OFFDRB, and process only U.S. office locations.
- 6) Call procedure FmtSndRateUpdts (module AML832A005) to send rates for the car classes retrieved. These will be written to AMSEND in X832F3 formats. If this is the initial load, specify business case '1'. If this is a yearly load, specify business case '3'.
- 7) If this is the initial load, call subprocedure SndOfcUpdt to send the office location information. This will be written to AMSEND in X832F2 formats using business case '1'.
- 8) Call procedures in module AML832A006 to build the key and write the end formats to file AMSEND.
- 9) Send a data queue to DQ832S containing the key with the transmission ID to initiate sending the information to the insurance customer when batch program AMZ832A runs.

**@Files: (CRUD)**

OFFDBB (-R--)  
ARMSPR1 (-R--)  
ARMSPR3 (-R--)  
MACHID (-R--)  
AMSEND (C---)

**Process**

Hierarchical numeric ID: 1.1.4

Coded name:

Name: AA ARMS Support

Comment: @Definition: The application area responsible for supporting ARMS communications environment, producing reports for customers and handling any operational problems that are discovered.

**Process**

Hierarchical numeric ID: 1.1.4.1

ARMS Process Report
---------------------

## Coded name:

Name: SA Manage Environment

Comment: @Definition: The activity that initiates the ARMS application and performs a controlled shutdown when needed. It also involves tasks to handle transactions bound for a machine that is currently unavailable.

## Process

Hierarchical numeric ID: 1.1.4.1.1

## Coded name:

Name: BAT Purge Application Database

Comment: @Definition: This activity archives and reorganizes the application database, including some of the transactions work files.

## Process

Hierarchical numeric ID: 1.1.4.1.1.1

## Coded name: AML992

Name: PGM Purge Cross-Reference and Associated Database File(s) by Vendor Id (AML992)

## Comment: @Purpose:

To maintain the ARMS database by purging (deleting) old records.

## @Operational Method:

- Calculate the oldest date for retained data.
- For each cross-reference record:
  - Delete all records, in the database, associated to the cross-reference where the status code and date criteria are met.
- Delete records as above until a shutdown request is detected.

@Notes: files AM095P and AM096P are retained for 25 months.

## @Files: CRUD

AMXREF	(-R-D)
AMNITLOG	(-R-D)
AMAUTD	(-R-D)
AMADJD	(-R-D)
AMCOMD	(-R-D)
AMINSO	(-R-D)
AMIEBH	(-R-D)
AMIEBD	(-R-D)
AMIEBT	(-R-D)
AMRATD	(-R-D)
AMRMTD	(-R-D)
AMRNTD	(-R-D)
AMRPRD	(-R-D)
AMSURD	(-R-D)
AMTIME	(-R-D)
AMMSCLOG	(-R-D)
AM990P	(-RUD)
AM095P	(-R-D)
AM096P	(-R-D)

## Process

<b>ARMS Process Report</b>
----------------------------

Hierarchical numeric ID: 1.1.4.1.1.3

Coded name: AML994A

Name: PGM Stop Cross-Reference and Associated Database File(s) Purge (AML994A)

Comment: @Purpose:

To change the control data area (AM006A) for the purge (program AML992A), instructing it to shutdown.

@Operational Method:

- Change the contents (position 1 - 8) of the data area to cause the program to shutdown.

#### Process

Hierarchical numeric ID: 1.1.4.1.1.4

Coded name: AML993A

Name: PGM Purge Orphan Transaction / Database Records (AML993A)

Comment: @Purpose:

To delete every record in the ARMS database, that does not have a corresponding cross-reference record.

@Operational Method:

- For each database file:
  - Read the file, deleting any record that does not have an associated cross-reference record.

@Note:

- Files AM0060SV, AMXFRLOG, and AMPGMERR are purged using a date obtained from the control file. Files AM095P and AM096P are retained for 25 months.

@Files

- AMERRHST	(-R-D)
- AM990P	(-R--)
AM0060SV	(-R-D)
AMADJD	(-R-D)
AMAUTD	(-R-D)
AMBILERR	(-R-D)
AMBILLOG	(-R-D)
AMCOMD	(-R-D)
AMIEBD	(-R-D)
AMIEBH	(-R-D)
AMIEBT	(-R-D)
AMINSD	(-R-D)
AMMNTLOG	(-R-D)
AMMSCLOG	(-R-D)
AMPGMERR	(-R-D)
AMRATD	(-R-D)
AMRCVERR	(-R-D)
AMRMTD	(-R-D)
AMRNTD	(-R-D)
AMRPRD	(-R-D)
AMSNDLOG	(-R-D)
AMSTSX	(-R-D)
AMSURD	(-R-D)
AMTIME	(-R-D)

ARMS Process Report
---------------------

AMTRNLOG	(-R-D)
AMXFRLOG	(-R-D)
AMXREF	(-R-D)
AM095P	(-R-D)
AM096P	(-R-D)

**Process**

Hierarchical numeric ID: 1.1.4.1.1.5  
Coded name: AML995A  
Name: PGM Stop Orphan Transaction / Database Records Purge (AML995A)  
Comment: @Purpose:

To change the control data area (AM008A) for the purge (program AML993A), instructing it to shutdown.

**@Operational Method:**

- Change the contents (position 1 - 8) of the data area to cause the program to shutdown.

**Process**

Hierarchical numeric ID: 1.1.4.1.1.6  
Coded name: AML996A  
Name: PGM Reorganized Application Database File(s) (AML996A)  
Comment: @Purpose:

To maintain the size of the database files by reorganizing the objects, thus reducing amount of time required when accessing the data.

**@Operational Method:**

- Update the controlling data area with the file name to be reorganized. That way when the job ends in mid stream it can pick up where it let off.

**Process**

Hierarchical numeric ID: 1.1.4.1.1.7  
Coded name: AML997A  
Name: PGM Archive Transaction File Data (AML997A)  
Comment: @Purpose:

Based on a practical experience of the Application Support Staff, data within certain transaction files are archived for trouble shooting purpose.

**@Operational Method:**

- Create the archive files when they don't exist for the host machine running the program.

- Each file follows the same archiving procedure:
  - Second month rolled to temporary Third month
  - First month rolled to Second month
  - Current month rolled to First month
  - Empty Third month inserted as Current month

ARMS Process Report
---------------------

@Note: Presently the data is kept for two months.

#### Process

Hierarchical numeric ID: 1.1.4.1.2

Coded name:

Name: BAT Startup ARMS Environment

Comment: @Definition: This activity reorganizes the transactions work files, data queues, archives needed information and prepares the environment for starting the ARMS application.

@Notes: The ARMS rejected transactions report is also generated by this activity.

#### Process

Hierarchical numeric ID: 1.1.4.1.2.1

Coded name: CLL813

Name: PGM Delete and Rebuild Data Queues (CLL813)

Comment: @Purpose:

To delete and rebuild data queues. This is an IBM recommended procedure to improve data queue performance.

#### @Operational Method:

- CALL CLL813HIST to archive file, AMSET.
- CALL AM0038V1 to save data queue entries (for all data queues except DQAM62V1) and associated data to work files for restoring at a later time
- CALL AM0037V1 to save AMPACK (ARMS Transactions ready to be sent) records that have not been processed to a work file for restoring at a later time
- Read records from ANDQINFO file in a loop and for each data queue in the file, perform the following steps:
  - i) IF data queue name is DQAM62V1 and the machine is NOT DEV, skip this record and read the next record ELSE continue
  - ii) IF the associated data queue file name is blank, skip this record and read the next record ELSE continue
  - iii) IF the machine is RARMS and "RARMS Only" flag is 'Y'es,
    - OR IF the machine is DEV
    - OR IF the machine is NOT RARMS AND the RARMS only flag is NOT 'Y'es
- IF the data queue name is blank, reorganize the data queue file
  - IF the VAN is not specified
  - OR IF the VAN is specified AND this is the weekly run, clear the data queue file.
  - IF the file is in use, message ARMS On Call
- IF on the DEV machine, CALL CLL814 (as it is not submitted by ROBOT on DEV). On DEV CLL813 is called by CLL991.

#### Process

Hierarchical numeric ID: 1.1.4.1.2.2

Coded name: CLL813HIST

Name: PGM Backup ARMS Critical Files (CLL813HIST)



<b>ARMS Process Report</b>
----------------------------

Comment:     @Purpose:

To backup files ANDIST, AMSET and AM0060LG.

@Operational Method:

- On the DEV machine, archive all three files with the \*REPLACE option on the CPYF command to keep the archived file size small.

- On the RARMS machine, archive the set file, AMSET with the \*ADD option to append to the previous archive

- On the distributed machine, archive the files ANDIST and AM0060LG to ANDIST\_HST and AM0060SV respectively with the \*ADD option to append to the previous archives.

### Process

Hierarchical numeric ID:       1.1.4.1.2.4

Coded name: AM0038

Name: PGM Save / Restore Data Queue Entries and Associated Data (AM0038)

Comment:     @Purpose:

To save the data queue's data to a work file and then delete and recreate the data queue to improve its performance.

@Operational Method: Three parameters are passed to this program: the processing code ('S'ave or 'R'estore), the data queue name and the run type ('D'aily or 'W'EEKLY).

For 'S'ave processing do the following:

- If the data queue name parameter is \*ALL, read file, ANDQINFO in a loop  
Else read the file with the data queue name as the key and for each data queue that meets the following conditions :

is not DQAM62V1

is not blanks,

the VAN id is blanks OR

the VAN id is not blanks and the run type is 'W'EEKLY,

the machine is RARMS and the "RARMS Only" flag is 'Y'es,

the machine is 'DEV'

the machine is distributed and the RARMS Only flag is NOT 'Y'es;

Do the following:

IF the key length is greater than zero

receive all keys greater than blank

ELSE

simply receive each entry sequentially.

For each entry except the \*DOWN, \*STOP and \*WAIT entries, write the data queue key and data to a work file until all entries have been received.

For each entry that has associated records in a database file, write the records to a work file.

Then delete the data queue and recreate it.

For 'R'estore processing:

ARMS Process Report
---------------------

- If the data queue name parameter is \*ALL, read the work file in a loop ELSE read the work file with the data queue name as the key and for each data queue entry found in this file,

- i) Read the file, ANDQINFO to get the data queue information
- ii) Send the queue entry to the corresponding data queue via one of the two subroutines, one for keyed and one for non-keyed data queues.
- iii) Write any associated database records in the work file back to the appropriate database file.

For any chain failing in the above, CALL AMPSSR and message ARMS On Call.

#### Process

Hierarchical numeric ID: 1.1.4.1.2.5

Coded name: AM0037V1

Name: PGM Backup Unprocessed Transactions in Package file AMPACK (AM0037V1)

Comment: @Purpose:

To save unprocessed AMPACK records.

@Operational Method:

Job L813 creates a work file, L813A by sorting AMPACK by company id.

Read previous L813A starting from the last record and save all 'IN'voice and 'OF'fice batch transactions that have not been sent.

Any company that has a format 'APPD02' record in AMPACK has already been processed and so do not save that company's data. Look for \*EOBD in the data portion and the format name 'APPD02'. This format is written by program AM0140 (called in CLL811) for each company that the batch transmission is sent to.

#### Process

Hierarchical numeric ID: 1.1.4.1.2.6

Coded name: CLL810

Name: PGM ARMS Startup (CLL810)

Comment: @Purpose:

To start up the ARMS application system programs that make the ARMS system handle transactions on a continuous basis until a shutdown is initiated.

@Operational Method:

- Call program CC0006 to reset the callback control status flags from "OK" to "DO" in CCEXTCTL Claims Connection Callback file.

- Reset the Number of Transmission Envelope Validation NEP jobs currently active to zero.

- Allocate file AMXREF to determine if there is lock on this file. If there is no lock DeAllocate file and continue. If the file is locked and not available delay job and try again. If on the third attempt the file is still locked page the ARMS oncall.

- Submit the proper Never-Ending Program (NEP) batch jobs to the ARMS Job queue for the centralized ARMS external transaction-based ("real-time") communications (separate sending and receiving jobs for each specific connection),

**ARMS Process Report**

centralized ARMS processing, centralized internal host platform distribution communications, distributed ARMS processing, distributed rental interface and distributed rental extraction transaction set generation processes. (This is based on the current host system name that this startup processing is being executed.)

**@Notes:**

This processing is initiated daily around 3:30am. Shutdown: 11:50pm  
(all times for Central Time Zone)

**@Files** (CRUD)

none

**Process**

Hierarchical numeric ID: 1.1.4.1.2.8

Coded name: CLL817

Name: PGM Copy / Print Rejected Transactions Report & Start Subsystem (CLL817)

Comment: @Purpose:

To consolidate all the transactions rejected by ARMS edits and generate an rejected transactions report after archiving the error log file, AMERRLOG. Also, start the ARMS subsystem if not already started.

**@Operational Method:**

- IF on the distributed machine, copy file AMERRLOG to the same file on the ARMS host via DDM and then clear the file. If DDM link is unavailable, message on call staff.

- IF on the ARMS host, archive consolidated file AMERRLOG and generate rejected transactions report by calling program AM5000. Reorganize file AMERRLOG to remove deleted records.

- Start the ARMS subsystem if not already started.

**Process**

Hierarchical numeric ID: 1.1.4.1.2.9

Coded name: AM5000

Name: PGM Print ARMS Error Log Report (AM5000)

Comment: @Purpose:

To generate the report for all the inbound and outbound transactions that were rejected by ARMS edits.

**@Operational Method:**

- Read file AMERRLOG sequentially. The file is keyed by Company Id, Transmission Control Id, Group Control Id, Group Type and Customer Transaction Id.

- For every record in this file, write the header and all error codes and associated detail. For error codes 68 (Invalid Vendor Transaction Id) and 82 (Request Rejected : can not identify transaction) on an Authorization (AT) group type, retrieve the SET 9formats beginning with "SET") and Renter (formats beginning with "RNT") detail formats from the received transactions log, AMTRNLOG and write them to the detail portion of the report.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.1.2.11  
Coded name: AM0032V1  
Name: PGM Write Saved AMPACK records to AMPACK (AM0032V1)  
Comment: @Purpose:

To restore the saved AMPACK records.

@Operational Method:

This program starts at the bottom of the work file where AMPACK records were saved and does a read prior to restore the data to the file, AMPACK in the proper sequence.

**Process**

Hierarchical numeric ID: 1.1.4.1.2.12  
Coded name: AML002A  
Name: PGM File Transfer and Archive RMS ARMS Trading Partner Insurance Company Transaction File (AML002A)  
Comment: @Purpose:

To transfer the daily RMS (Rental Management System) transaction file record on the centralized ARMS host platform to the RMS interface file on the RMS centralized host platform.

@Operational Method:

This program is submitted to run as a batch job on the centralized ARMS host platform after the CLL814 program batch job has completed and before the CLL810 program batch job has been submitted. The program will attempt to allocate exclusively and if successful, then copy the ARMS RMS Daily Transaction file (AM002P) to an FTP (File Transfer Protocol) remote site and an archive data set.

Retrieve the current platform (machine) name.

IF the platform name is not 'RARMS' (centralized ARMS host), or 'CENTRAL' (centralized RMS host), or 'DEV' (IS development host), then end this program/job.

Retrieve the current executing job's SUBSYSTEM DESCRIPTION name, JOBNAME and JOBDAT.

Clear the RMS staging area file (AM002PFTP).

IF the current platform is RARMS or DEV, then do the following:

.. Check for an empty RMS Daily Transactions (AM002P) file.  
.. IF the RMS Daily Transactions (AM002P) file is empty (no records), then end program/job.  
.. ELSE, Attempt to Allocate file AM002P to prevent all other access.  
....IF unable to allocate exclusively the AM002P file, then end the program and send a corresponding Enterprise Message System message to the ARMS On-call staff along with a pager message to the ARMS Primary and Backup On-Call pagers indicating "AM002P File is in use".  
.. AND: Copy the AM002P file to the currently empty RMS interface staging area file (AM002PFTP) sequentially from the beginning of the file.  
.. AND: Clear and Deallocate the AM002P file, making it reusable.

## ARMS Process Report

IF the current platform is CENTRAL or DEV, then do the following:

..IF RARMS is remotely unavailable ("platform or intersystem communications link is down"), then send a corresponding message of the remote communications failure to the current platform's system operator (QSYSOPR) and resubmit this job on hold for execution later on this platform, using the previously retrieved JOBNAME, JOBDATE and SUBSYSTEM DESCRIPTION name as the JOB QUEUE name and then end this program/job.

..ELSE (RARMS platform is remotely available), then do the following:

....Perform a "copy-pull" with replacement by copying the records sequentially from the AM002PFTP file on centralized RARMS platform to the current platform's AM002PFTP file.

....Check for an empty RMS Daily Transactions Interface Staging Area (AM002PFTP) file.

....IF the RMS Daily Transactions Interface Staging Area (AM002PFTP) file is empty (no records), then end program/job.

....ELSE, do the following:

.....Retrieve the 4 character day of week abbreviation (current OS/400 system value) (Example: 'THUR', 'WED ', 'TUES')

.....Append the first 3 character of the 4-character day of week to the 'AM002P' file name to derive the correct archive file for each day of the week.

.....Copy (sequentially with replace) the file to be transferred via FTP to the derived archive file for today's day of the week.

.....Retrieve the AM007A data area to get the Local Area Network FTP site name for RMS download. (Example: 'FTPCORP01')

.....Execute the FTP (File Transfer Protocol for AS/400 to Personal Computer) for downloading using the QTXTSRC Text Source File's AMRMSFTPO member for the File Transfer Input Specifications to transfer/copy with append the records on the AS/400 platform in the RMS Daily Transaction Interface Staging Area (AM002PFTP) file into the AM002P.TXT file in the "\rms\" directory on the FTPCORP01 LAN server's public drive.

@Files: (CRUD)

AM002P (-R-D)

AM002PFTP (CR--)

AM002P.TXT (C---) (in the \rms directory of the FTPCORP01 specified drive of the LAN server's public drive)

@Improvement Opportunities:

Change this process from a batch process to a real-time process to send/push any qualified transaction into the RMS database files.

### Process

Hierarchical numeric ID: 1.1.4.1.2.13

Coded name: CLL814

Name: PGM Restore Unprocessed Transactions (CLL814)

Comment: @Purpose:

To restore the data queue entries and associated file data that was saved in job L813.

## ARMS Process Report

### @Operational Method:

The run type 'D'aily or 'W'weekly is passed to this program as a parameter.

- CALL AM0032V1 to restore file AMPACK
- CALL AM0038V1 with the processing code set to 'R'estore to restore the data queues (these data queues contain the key and the data in the data queue itself e.g. DQAM60V1)
- Save the work files and then clear them

### Process

Hierarchical numeric ID: 1.1.4.1.3

Coded name:

Name: BAT Shutdown ARMS Environment

Comment: @Definition: This activity shuts down all the ARMS never ending programs that constitute the main ARMS transaction processing stream.

@Notes: The transactions processing stream consists of the following:

- Translators (AM10)
- Host Processing (AM20/21-AM46)
- Distributed Sending (AM100-AM101) from host
- Distributed Inbound Processing (AM61-EC(AT,EX,CN,CM,CC))
- Distributed Outbound Processing (AM60-EC(AC,RA,RE,RC,RN,VM,TR))
- Distributed CC Inbound Processing (AM61-CC(AT,EX,CN,CC))
- Distributed CC Outbound Processing (AM60-CC(AC,RA,RE,RC,RN,IN,TR))
- Distributed Receiving (AM105-AM106) to host
- Host Sending (AM120)

The receiver jobs remain active 24 X 7 hours except when a roll swap is done to switch all processing to the backup ARMS host machine.

The first shutdown entry is sent to each of the translators running (via program AMSTOP2X which is called from CLL812 which is a scheduled job) so that all input to the stream can be stopped. The translator decrements data area DASDV1, passes shutdown to AM20 and ends itself. AM20 monitors data area DASDV1 and when it becomes zero, it passes the shutdown to AM21 and ends itself. AM21 in turn passes the entry to AM25 and ends itself. AM25 then reads data area AM001A to get the number of computer platforms to shutdown and sets on an internal flag to indicate to itself that one shutdown entry has already been received. After this AM25 passes the entry to AM30 and so on up to AM46. These programs do not end with the first shutdown entry in order that they be able to process any data coming from the distributed computer platforms.

AM46 then passes the entry to each of the distributed sending jobs (AM100) so that the distributed streams can be stopped. AM100 passes the shutdown to AM101 with a detach signal. When AM61 receives the shutdown entry, it reads file AMVRGTST, keyed by group type and source type (e.g. ATB, ATC etc.) to shutdown each of the inbound format generators and then sends the shutdown to AM60 which in turn stops the outbound format generators before ending itself. Each of the format generators, AM61 and AM60 send a shutdown entry to AM106 before ending. AM106 maintains a count of all applicable records in AMVRGTST and when that count is met, it passes the shutdown entry to AM105 along with a detach signal before ending.

Note that AM101 and AM105 can distinguish between a single machine shutdown and a complete system shutdown. For e.g., AM105 will not pass the shutdown to AM25 if it was a single machine shutdown.

**ARMS Process Report**

AM25 then increments the data area, AM002A, that contains the count of computer platforms that have shutdown. When this count equals the machine count in data area AM001A (manually maintained), AM25 knows that the required number of computer platforms are down and now it can end the remaining host processing stream that is still up by passing on the entry until it reaches AM46.

AM46 then passes the entry to AM120 which stops each of the senders based on the file ARMSPR3 and then ends itself.

**Process**

Hierarchical numeric ID: 1.1.4.1.3.1  
Coded name: CLL812  
Name: PGM Stop ARMS Never-Ending Programs (CLL812)  
Comment: @Purpose:

To initiate the automated and orderly process for ARMS system shutdown in order to completely process any transactions that have already been started and then shutdown the whole system.

**@Operational Method:**

- Send Shutdown Request transaction to Trading Partners so profiled to be able to receive a shutdown request.

- After five (5) minutes, then end the remaining communication jobs via an ENDJOB command.

- Send a Shutdown Transaction to each connect-specific receiver output data queue.

**@Notes:**

Upon the receipt of the first shutdown transaction by the centralized validation activity component programs (AM0025, AM0030 and AM0040) and the centralized database update activity component process program (AM0046), the first shutdown transaction is indicated and the shutdown transaction is passed along to the next program. After receiving the second shutdown transaction, the shutdown transaction is passed to the next program and causes the current program to end in an orderly manner.

**Process**

Hierarchical numeric ID: 1.1.4.1.3.2  
Coded name: AMSTOP2X  
Name: PGM Stop All Active Translator jobs (AMSTOP2X)  
Comment: @Purpose:

To shutdown all translator and any receive mapper jobs running in the ARMS subsystem.

**@Operational Method:**

- Retrieve all companies and technical connection information for each company from the profile file, ARMSPR3 and send a shutdown entry to the appropriate receive data queue to shutdown the translator (AM10) job for each company or VAN as the case maybe.

**ARMS Process Report**

- Then send a shutdown entry to the ARMS Trading Partner Insurance Company receive mapper job. This is hard coded in this program.

**Process**

Hierarchical numeric ID: 1.1.4.1.3.3  
Coded name: RSWAPSTR  
Name: PGM End ARMS jobs if necessary (RSWAPSTR)  
Comment: @Purpose:

To shutdown the ARMS processing stream on the current machine as part of the preparation for the role reversal between the current and the backup machine.

**@Operational Method:**

- Check the ARMS and A4BAT subsystems for jobs to see if jobs L812 and LAMSTOP2R have run.

If there are jobs running in either subsystem do the following:

- i) if the time is between 0300 and 2345, submit a job to perform an immediate end of the ARMS processing stream by calling CLL812 with a delay of zero.
- ii) if the time is between 0300 and 2200, end jobs L811 and L749.
- iii) if the time is between 2245 and 2530, call AMSTOP2R with a zero delay to immediately end all the receiver communication jobs.
- iv) once all the above steps have been performed, end the ARMS subsystem to flush any remaining jobs.

**Process**

Hierarchical numeric ID: 1.1.4.1.3.4  
Coded name: CLSR01  
Name: PGM Role reversal CLP (RARMS1 <-> RARMS2) (CLSR01)  
Comment: @Purpose:

To shutdown all processing on the current machine and start the ARMS processing on the backup machine.

**@Operational Method:**

- This CL is executed by operations when there is a need to perform a role reversal between the current and the backup ARMS machine.

**@Notes:**

This CL includes a lot of processing for ending several subsystems that may be running on the current machine and the code is maintained by operations.

The source for this CL can be found in the SPLIB/QCLSRC.

**Process**

Hierarchical numeric ID: 1.1.4.1.3.5  
Coded name: AMSTOP2R  
Name: PGM Stop All Active Comm. Receive Jobs (AMSTOP2R)



**ARMS Process Report****Comment: @Purpose:**

To shutdown the communications receiver jobs for all companies and to end the pseudo sender job, A4200RB for the special case of ARMS/400.

**@Operational Method:**

- Retrieve the profile information to construct the receiver job name for each company and end the job.

- For ARMS/400, send a data queue entry to DQ4200RB to end the A4200RB (ARMS/400 pseudo sender) NEP.

**Process****Hierarchical numeric ID:** 1.1.4.1.3.6**Coded name:** DQ4200RB**Name:** DTQ Input for ARMS/400 Application (DQ4200RB)**Comment: @Definition:** This is the input data queue to the pseudo sender for ARMS/400, program A4200RB that simulates sending to the ARMS/400 customers.**Process****Hierarchical numeric ID:** 1.1.4.1.4**Coded name:****Name:** ONL Reroute Unsolicited Authorizations**Comment: @Definition:**

To activate the rerouter job for any unavailable machine.

**@Operational Method:**

When a production machine goes down, the ARMS OnCall staff is notified and then the OnCall staff manually submits this job to reroute or hold any transactions that were meant for that machine.

**@Note:**

- Runs on RARMS only

**Process****Hierarchical numeric ID:** 1.1.4.1.4.1**Coded name:** AMRREROUTE**Name:** PGM Submit ARMS Rerouter (AMRREROUTE)**Comment: @Definition:**

This is a batch process triggered manually as a result of a distributed production machine that has been abnormally shut down to allow for certain transactions that would have been routed to that machine to instead be routed to CLAIMS CONNECTION and the remainder of the transactions intended for that machine to be held for later processing.

**@Purpose:**

To allow Enterprise a means of fulfilling contractual ARMS obligations in event of a distributed machine failure.

<b>ARMS Process Report</b>
----------------------------

**@Operational Method:**

- Submits the AMRREROUT never-ending program with all of the necessary values.

@Files: None

**Process**

Hierarchical numeric ID: 1.1.4.1.4.2

Coded name: AMRREROUT

Name: PGM Reroute Transactions for Unavailable Machine (AMRREROUT)

Comment: @Purpose:

To handle transactions that are bound for a machine that is unavailable so that the transactions are not lost. If the transactions are new authorizations, someone needs to attend to them as most trading partners have agreements with Enterprise to respond to customer authorization requests within a specified time interval.

**@Operational Method:**

IF the received transaction set is an unsolicited authorization add (AT);

- Call AM1010V1 to update the XREF with Machine & Source Id, update the AMAPP record and route to Claims Connection

IF the received transaction set is an authorization change (AT-C) OR a cancellation (CN), and the Source or Machine from the APPD01 doesn't match that found in the XREF;

- update the AMAPP record and route the transaction to Claims

**Connection**

ELSE

- route to hold data queue.

At SHUTDOWN, reroute all held transactions to the production machine that went down.

**@Files:**

- AMAPPS (-RU-)

- MACHIDENPF (-R--)

**Process**

Hierarchical numeric ID: 1.1.4.1.4.3

Coded name: DQAMAPH

Name: DTQ Hold Unavailable Machine Transactions (DQAMAPH)

Comment: @Definition: DQAMAPH is a data queue used during reroute for holding all transactions that are not unsolicited AT-Adds (which are routed to CLAIMS CONNECTION) for the affected machine which is down, and then at reroute shutdown for the that machine these entries are then used as input to allow for their normal processing.

**Process**

Hierarchical numeric ID: 1.1.4.2

Coded name:

Name: SA Research and Fix Problems

**ARMS Process Report**

Comment: @Definition: The activity to handle any problems that are reported via a page, a phone call or an MS01 message. It also involves daily tasks to examine exceptions and fix them or to create issues if they can not be fixed.

**Process**

Hierarchical numeric ID: 1.1.4.2.1

Coded name:

Name: MAN Respond to On-Call Message

Comment: @Purpose:

To research and determine how to solve a problem that was notified via a pager, a phone call or an MS01 message.

**@Operational Method:**

- The On-Call staff (primary and backup) are notified in the following situations:

1) ARMS job's program halt or termination via pager message to investigate why and to respond and possibly resubmit job for never-ending program (NEP) execution.

2) ARMS transaction set transmission being rejected for processing by an ARMS NEP job via a pager message and/or via e-mail (currently MS01).

- The primary only On-Call staff is notified by ARMS Rental Management Trading Partner technical contact of their host system availability or connect problems or for transaction research for a perceived problem via pager message (8063). The contact actually dials in (314) 512-ARMS and is appropriately directed via four options :

- 1 - Enterprise Rental Branch Personnel
- 2 - Questions about Vehicle Messaging System
- 3 - Insurance Company Contacts
- 4 - Other

The primary only On-Call staff is also notified by the ARMS Programming Help Desk staff via pager message (8064) of problems where they need further assistance.

- To retrieve the 8063 and 8064 mail box messages, perform the following steps:

- i) Dial phonemail - x2696
- ii) When asked, enter the mailbox number from the pager (8063 or 8064) and the password
- iii) After listening to the message, always save it
- iv) Call the concerned party and always keep them apprised of the status of the investigation.

**@Notes:**

A top priority is to resubmit/restart any abended ARMS NEP job to ensure transaction data flow throughput between the trading partners and the rental systems. Another high priority is to notify ARMS Technical to contact all affected trading partners of any communication termination occurrences that are estimated to be longer than thirty (30) minutes in duration. Likewise, they are also to be notified of any similar delays of transaction throughput. A lower priority is to research the cause of any transaction being rejected or causing program halt/termination, correct and resubmit the transaction.

<b>ARMS Process Report</b>
----------------------------

**Process**

Hierarchical numeric ID: 1.1.4.2.2  
Coded name:  
Name: MAN Perform Daily Exception Checks (Checklist)  
Comment: @Purpose:

To use the check list and perform daily tasks that have been identified as routine to make sure that the application processes any exceptional transactions.

**@Operational Method:**

- If any printed detail lines exist on the ARMS Cross-Reference Discrepancy Reports spooled file in the ARMSRPTS output queue on the RARMS centralized host platform, print the reports. Then research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed group type.

- Display the RARMS host platforms distribution data queue (DQAMAP1) to determine if there is any old (older than today's date) or hung transaction that has never been sent. Research the associated transaction set's transmission and data queue entry for invalid host system platform ID value. Correct and reprocess if necessary.

- Display the distributed rental systems' host platforms ARMS Application Interface Transaction Set file (AMAPP), the ARMS Format Generation Dispatcher In-Progress Transaction Control File (AMTRNCTL), and all distributed ARMS application system data queues (DQAM6AV1, DQAM6BV1, DQAM6OV1, DQAM61V1, and DQECATV1) to determine if there is any old (older than today's date) or hung transaction that has never been processed. Research the associated transaction set's transmission and data queue entry (along with the existing rental contract transaction) for invalid rental contract IDs values or status. Correct and reprocess if necessary.

- Check the spooled print file archive system's spooled file for the latest J340X Daily Rental Billing Job runs' Billing Exception Report for rejects due to ARMS status code or missing ARMS Cross-Reference file record error message. If any exist, note the rental location and contract ID. Research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed group type.

- Check if any printed detail lines exist on the query reports from last execution of job AML900 in two spooled files in the ARMSRPTS output queue on the RARMS centralized host platform, print the reports. Then research all ARMS activity to-date of the rental transaction via the ARMS Time-Line Inquiry (CCAM16) and the ARMS centralized database report (AM0800). Manually correct the files' records or generate the necessary transaction set for the needed missing opening "RN" group type.

- Every weekday morning, change the centralized Job Monitor switch (on CENTRAL) for the ARMS On-Call Pager to "YES" so that ARMS On-Call pagers will be immediately sent notification messages of any ARMS job halts or abends.

- Every business weekday afternoon around 4:00pm, except the last business day of the week, change the centralized Job Monitor switch (on CENTRAL) for the ARMS On-Call Pager to "NO" so that ARMS On-Call pagers will NOT be sent

**ARMS Process Report**

notification messages of any ARMS job halts or abends. These notification pager messages will be sent instead to the Operations On-Site Pager.

**Process**

Hierarchical numeric ID: 1.1.4.2.3

Coded name:

Name: MAN Research and Resolve Network Problem Ticket

Comment: @Purpose:

To research and resolve and problem for which a network ticket has been created by the technical support center.

@Operational Method:

- Any time a Network Problem Ticket is passed to the Technical Support Center's Programming Help Desk (TSC-PHD) staff require further research and/or suggestions on correcting a deficient transaction situation with a rental transaction that had been processed through ARMS, then they call the ARMS Technical Support Line that generates an "8064" mailbox pager message to indicate the Application Development Department primary on-call person that a new voice mail message has been placed into the "8063" mailbox. The on-call person will listen to the voice mail message and then contact the caller to discover more details that might aid in their research. Then the on-call person (or can pass the task to the backup on-call if the primary has no time available for such a task) does the research into the problem and reports back to the caller the necessary intervention that the on-call person, TSC-PHD person, Rental Systems Help Desk staff, rental user, or trading partner system user must perform to correct the deficient situation.

**Process**

Hierarchical numeric ID: 1.1.4.3

Coded name:

Name: SA Information System Reports

Comment: @Definition: The activity that prints the various reports provided for in the ARMS application. The reports are as follows:

- Insurance company referral closed rental contracts revenue
- Management report for direct billed invoices
- Month-end ARMS youthful drivers surcharge discount report
- ARMS transaction exception report
- Reservation detail by company utilization management report

**Process**

Hierarchical numeric ID: 1.1.4.3.1

Coded name: CLJ470

Name: BAT Print Insurance Company Referral Closed Rental Contracts Revenue Reports (CLJ470)

Comment: @Definition: The batch process for printing of various Insurance reports, as specified in the Operational method and run on the Query machine. This process is triggered by TIME.

@Purpose: To report on various ARMS and Rental data in support of the Marketing function.

@Operational Method:

<b>ARMS Process Report</b>
----------------------------

- Print Insurance Referral Report of Gross Revenue
- Print Insurance Referral Report of Income
- Print Bill-To Customer Report by Group
- Print Referral Reports by State (Gross and Net Revenue)
- Print Bill-To Customer Report by State

@Files: (CRUD) (Unless noted, all files used reside on QUERY system)

- OFFDRB (-R--)
- IR470P (-R--) (on CENTRAL system)
- INSRMOYR (-RU-) (on CENTRAL system)
- INSRMOYR (CRU-)
- INSRGGP (-R--) (on CENTRAL system)
- INSRGGP (CR-D)
- INSRPHD (-R--) (on CENTRAL system)
- INSRPHD (CR-D)
- RACCLSC (-R--)
- RACCLS01 (-R--)
- RACCLS02 (-R--)
- RACCLS03 (-R--)
- RACCLS04 (-R--)
- RACCLS05 (-R--)
- RACCLS06 (-R--)
- RACCLS07 (-R--)
- RACCLS08 (-R--)
- RACCLS09 (-R--)
- RACCLS10 (-R--)
- RACCLS11 (-R--)
- RACCLS12 (-R--)

@Notes:

These reports are confidential and are used by the Rental Operations - National Marketing Department staff.

Process

Hierarchical numeric ID: 1.1.4.3.2

Coded name:

Name: BAT Transaction Credit Discount

Comment: @Definition: The batch process for printing the month end transaction credit report.

@Purpose: For ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS utilization for any transactions credit rebate checks sent to the Rental Management Trading Partner from closed direct billing payment authorized rental contracts.

@Notes:

Report is used by ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS usage for any transactions credit rebate checks sent to the Rental Management Trading Partner.

<b>ARMS Process Report</b>
----------------------------

**Process**

Hierarchical numeric ID: 1.1.4.3.2.1  
 Coded name: AML968A001  
 Name: PGM Create Missing OUTQs for Credit Transactions Report (AML968A001)  
 Comment: @Purpose:

To create missing output queues on the RARMS machine at the beginning of the Credit Transaction Report driver job (CLL968) program so it can complete normally.

These output queues are named: Pgg, PggPC, or PggRPC where "gg" is the Group ID and "r" is Region ID and are created in library QUSRSYS using the CRTOUTQ command with default parameters except for the OUTQ name.

**@Operational Method:**

Repetitively read through Office Directories Group Master File by GROUP ID (file OFFDG#2).

For every record read, do the following only IF the record's GROUP ID field value is not equal to '76' (Claims Connection) AND the GROUP ID field value is equal to the GROUP'S OUTPUT QUEUE NAME:

- IF the record's Distributed Printing Capable indication field is 'Y' (Yes - a PC printer output queue is located at this Group administration office), then check if that Group's PC output queue name ("PggPC" - where "gg" is the Group ID) exists on this platform. IF it does not exist, then create the missing personal computer (PC) output queue (OUTQs) for this Group Administrative office.

- ELSE (Yes - a PC printer output queue is NOT located at this Group administration office), then check if that Group's normal output queue ("Pgg" - where "gg" is the Group ID) exists on this platform. IF it does not exist, then create the missing normal peripheral output queue (OUTQs) for this Group Administrative office.

- IF the retrieved record's Regionalized indication field is 'Y' (Yes - this Group is administrated by other regional administration offices), then execute the ARMS Retrieve Regions for Office Group (AML968A002) program, passing two parameters. This first is the input parameter of the current record's GROUP ID. The second parameter is the blanked 132 element, 1 character output parameter of a Region ID array of all regions that are part of a Group office. Then, looping through each non-blank array element, check if the Regional Office's of the Group personal computer output queue name ("PggRPC" - where "gg" is the Group ID and "r" is the Region ID) exists on this platform. IF it does not exist, then create the missing personal computer (PC) output queue (OUTQs) for this Group's Regional Administrative office.

@Files: (CRUD)

OFFDRGRPPF (-R--) Office Directories Group Master File (by Group ID, using logical access path OFFDG#2).

**Process**

Hierarchical numeric ID: 1.1.4.3.2.2

<b>ARMS Process Report</b>
----------------------------

Coded name: AM1081V1

Name: PGM Print Detail and Summary Transaction Credit Reports (AM1081V1)

Comment: @Purpose:

To print ARMS Transaction Credit summary and detail reports by Trading Partner.

@Operational Method:

Positions and reads the ARMS Transaction Credit Transaction History (AMTRNCR) file records whose Year and Month Approved for Payment/Rental Closed correspond to the previous month/year from the current system date.

This physical file's indexed/keyed access path is sequenced by:

Year Approved for Payment or Rental Closed  
 Month Approved for Payment or Rental Closed  
 Trading Partner Profile ID  
 Group ID  
 Region ID  
 Branch ID  
 Ticket ID Number (of closed rental contract)

Regionalized the Transaction Credit summary report. This report matches the report that is sent to the Group output queues except that it includes total information for branches within a region and total information for the region itself, but does not include Group total information.

A total line to the Transaction Credit Detail report exist just for ARMS Trading Partner Insurance Company. The total will be the sum of the transaction credit and the utilization fee.

(If there is a Trading Partner with no utilization fee, this total line will not print.)

The hierarchy of breaks are:

Trading Partner Profile  
 Group  
 Region  
 Branch

Totals for: Accumulate Branch, Region (if applicable), Group and Trading Partner

Total Count of (Rental Contracts) Tickets Closed (Managed via ARMS)  
 Total Insurance Charged Extended Daily Rental Rate Revenue Amount of (Rental Contracts) Tickets Closed (Managed via ARMS)  
 Total Transaction Credit (Discount) Amount (to be rebated)  
 Total (ARMS) Estimated Utilization Fee Amount

Print detail report lines (used for auditing when necessary)

Print report break totals on detail and summary report printer files.

Page breaks on: Trading Partner Profile ID or GROUP ID changes.

@Files: (CRUD)

AMTRNCR (-R--) ARMS Transaction Credit transaction history file

ARMSPR7 (-R--) ARMS Trading Partner's Transaction Discount  
 Volume Break Control Levels



<b>ARMS Process Report</b>
----------------------------

ARMSPR2                    (-R--) ARMS Trading Partner's Customer Address Data

OFFDRGRPPF            (-R--) Office Directories Group Master File, by GROUP ID  
(using the OFFDG#2 logical access path)

QSYSPRT                (C---) System Default PRINTER FILE for Detail Report by  
GROUP ID, REGION ID, and BRANCH ID

QSYSPRT2              (C---) System Default PRINTER FILE for Summary Report by  
GROUP ID, REGION ID, and BRANCH ID

QSYSPRT3              (C---) System Default PRINTER FILE for Detail Report by  
GROUP ID, REGION ID, and BRANCH ID for the Group Administration Office Output Queue

QSYSPRT4              (C---) System Default PRINTER FILE for Detail Report by  
GROUP ID, REGION ID, and BRANCH ID for the Regional Administration Office Output  
Queue

@Embedded Data/Constants:

Program Constants:

'GROUP SUMMARY - FINAL TOTALS'  
'UTIL FEE \$'  
'UTIL FEE'  
'UTIL %'  
'TOTAL'

\*\* CMD - Commands To Be Executed

OVRPRTF FILE(QSYSPRT3) OUTQ(QPRINT)  
OVRPRTF FILE(QSYSPRT3) OUTQ(PggPC)  
OVRPRTF FILE(QSYSPRT3) OUTQ(Pgg)  
DLTOVR FILE(QSYSPRT3)  
OVRPRTF FILE(QSYSPRT4) OUTQ(PggrPC)  
DLTOVR FILE(QSYSPRT4)

\*\* Month Names for report column headings.

JANUARY  
FEBRUARY  
MARCH  
APRIL  
MAY  
JUNE  
JULY  
AUGUST  
SEPTEMBER  
OCTOBER  
NOVEMBER  
DECEMBER

@Notes:

Utilization fee is the amount paid by Enterprise to the customer for sending rentals using ARMS as specified in the contract with the company. For ARMS TRADING PARTNER INSURANCE COMPANY, currently they are paid a utilization fee of 1.8 % of any revenue generated via ARMS that is over 55 million.

The report is broken down by group and branch so that corporate can recover these fees from the appropriate group-branch based on the revenue that each branch generated via ARMS.

<b>ARMS Process Report</b>
----------------------------

**@Improvement Opportunities:**

Convert program from OPM to ILE.

Remove the compile-time array for the months' names and replace with the logic to execute a built-in function utility or service program that you can pass the date and the month name will be returned as output.

After having printed and saved the spooled printer files, if specified, then move the spooled files to another specific output queue for interception for archival by the Broderick Spool Archival utility software system. This would prevent the need for any reruns for the sake of a reprint.

Move the logic that provides the month and year values to select records for reporting from this program to the calling program, CLL968, and pass the derived last month's month and year as input parameters to this program. This would enable rerunning this program for a different past month that data was still retained on-line.

Optimum: Capture and retain all necessary information in a revised closed rental contracts file and use this type report over any closed rental contracts files by a past month and year for any rerunning.

**Process**

Hierarchical numeric ID: 1.1.4.3.2.3

Coded name: AML968A002

Name: PGM Retrieve Regions for Office Group (AML968A002)

Comment: @Purpose:

To return all associated Region IDs for a given Group ID so that the AML968A001 program can check if the regional administrative office's personal computer output queue exists on the current platform and if not existing, create them prior to the execution of the Transaction Credit Report (AM1080V1) program.

**@Operational Method:**

This program accepts two parameters, the first is the input 2 character GROUP ID. The second is the 132 element, 1 character each output REGION ID LIST array.

Starting at the first record in the Office Directories Branch Office Master File for the passed GROUP ID, read every record for this passed GROUP ID value until it changes or is end-of-file, do the following for each associated record retrieved:

- IF the REGION ID is NOT blank, then load the next available REGION ID LIST array element.

End the program and return the loaded REGION ID LIST array to the calling program.

@Files: (CRUD)

OFFDRB (-R--) Office Directories Branch Office Master File, by Group ID, Region ID, Area ID & BranchID (via OFFDRB7 logical access path).

**ARMS Process Report****@Improvement Opportunities:**

Condition loading the array only after checking to ensure that the retrieved OFFDRB7 file's non-blank REGION ID field value is different than the last loaded value, since the file access sequence is in GROUP ID and REGION ID ascending order. This will avoid loading the array with repeating values. Likewise, the 132 element REGION ID LIST array would not need the maximum of 132 and could be reduced to a smaller size such as the 94 characters (upper and lower case alphabetic, numeric, and special characters) that exist on the computer terminal's keyboard.

**Process**

Hierarchical numeric ID: 1.1.4.3.2.5  
Coded name: CLL968  
Name: PGM Drive Transaction Credit Discount Report (CLL968)  
Comment: @Purpose:

To print summary and detail reports, by Rental Management Trading Partner companies, of ARMS authorized transactions count and amounts that became closed authorized rental contracts and include the amount electronically invoice billed, excluding government taxes and surcharges.

These reports assist ARMS Technical Management and Rental Operations National Marketing Department staff to compute amount of ARMS utilization by Rental Management Trading Partner so that appropriate rebate checks may be sent to the trading partner.

**@Operational Method:**

Determine the current emulated or real hardware platform upon this program is executing.

IF the current hardware platform is "RARMS", then do the following:

Override the default system printer files QSYSPRT and QSYSPRT2 to have the attributes of COPIES set to 2.

Execute the ARMS Create Missing Output Queues for Credit Transactions Reports (AML001A001) program.

Execute the ARMS Print Detail and Summary Transaction Credit Reports (AM1080V1) program.

**@Notes:**

Submitted for execution from ROBOT/Scheduler as "L968C05" every month on the fifth calendar day of the month only on the RARMS system.

Report is used by ARMS Technical Management and Rental Operations National Marketing Department staff to compute ARMS usage for any transactions credit rebate checks sent to the Rental Management Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.4.3.3  
Coded name: CLL480  
Name: BAT Print Management Report for Direct Billed Invoices (CLL480)  
Comment: @Purpose:

**ARMS Process Report**

To report on various ARMS and Rental data relating to direct billed invoices in support of the Marketing function.

**@Operational Method:**

- Prints or sends to National Marketing representative (via FTP) the Rental Management Trading Partner Company Generic Insurance Direct Billed Report, for Selected Customers or a set of standard Branch Claims Office Customer Numbers.

**@Files: (CRUD)**

- INSRL480 (-R--)
- IR002P (CR-D)
- IR003P (CR-D)
- OFFDRB (-R--)
- RACCLS01 (-R--)
- RACCLS02 (-R--)
- RACCLS03 (-R--)
- RACCLS04 (-R--)
- RACCLS05 (-R--)
- RACCLS06 (-R--)
- RACCLS07 (-R--)
- RACCLS08 (-R--)
- RACCLS09 (-R--)
- RACCLS10 (-R--)
- RACCLS11 (-R--)
- RACCLS12 (-R--)

**@Notes:**

These reports are confidential and are used by the Rental Operations - National Marketing Department staff.

FTP process - Reports being downloaded are written to members in file IR002P. The names of these members are written as records in file IR003P. Program IR\_018M is then called by CL480 to send the data to the internet location. IR\_018M is made of 4 modules: CL IR\_018M001 which calls IR\_018M002 (BUILDER) to build the FTP copy commands (one per member) and to separate the members in IR002P into files in library QTEMP (module IR\_018M004 does this piece) and then calls IR\_018M003 (CHECKER) to read the FTP log to see if errors occurred. A batch job then runs on the internet location which checks for data in the appropriate directory and, if data exists, creates an executable that appears on Cindi Loomis of National Marketing's PC desktop. This batch job runs hourly Monday through Friday during normal business hours. There will be one executable for each downloaded file. When Cindi selects these, they are copied to diskette.

**Process**

Hierarchical numeric ID: 1.1.4.3.4

Coded name:

Name: BAT Youthful Drivers Surcharge Discount Report

Comment: @Purpose: The batch process of generating and printing the month end management report that deals with youthful drivers surcharge discounts

**@Notes:**

Distributed to Mary Mahoney, Rental Operations - National Marketing Department Assistant Vice President

ARMS Process Report
---------------------

## Process

Hierarchical numeric ID: 1.1.4.3.4.1

Coded name: AM1087V1

Name: PGM Generate Youthful Drivers Surcharge Discount Report (AM1087V1)

Comment: @Purpose:

To generate youthful drivers surcharge discount report.

@Operational Method:

- Print ARMS Youthful Driver Approved Surcharge Discount Detail Report by Group/Branch.

- Print ARMS Youthful Driver Approved Surcharge Discount Summary Report by Group.

- Print Exception Report of ARMS Transaction Credit Discount Records with closed rental contract (ticket) file record not found.

@Notes:

This report is used by National Marketing to determine discount benefit levels. Essentially, if the underage driver surcharge is not paid by the trading partner, it a discount for them and this report lists the actual discount amounts by trading partner.

This report is currently hard coded to be produced only for ARMS TRADING PARTNER INSURANCE COMPANY. The surcharge amount used for calculating the discount is \$5.00 per day per qualified rental. Any rental in AMTRNCR is qualified. The number of days for the rental is calculated from information retrieved from the closed ticket files on the QUERY machine.

## Process

Hierarchical numeric ID: 1.1.4.3.4.2

Coded name: CLL987

Name: PGM Setup Youthful Drivers Surcharge Discount Report (CLL987)

Comment: @Purpose:

To prepare the environment and run the program to generate the youthful drivers surcharge discount report.

@Operational Method:

- 1) Retrieve the current date and construct the REPORT DATE in format CYYMMDD using the previous month with the appropriate century and year.

- 2) Retrieve the machine name.

- If currently running on centralized ARMS host,

- retrieve the QUERY machine status

- If DDM is not available,

- Submit the job so that when the machine is available, the job can be released by operations

<b>ARMS Process Report</b>
----------------------------

Else

Return.

3) Change the printer file attributes so that 2 copies are generated for each report.

4) CALL program AML1087V1 with the REPORT DATE as the parameter.

5) Return

**Process**

Hierarchical numeric ID: 1.1.4.3.5

Coded name:

Name: BAT Cross-Reference Exception Reports

Comment: @Definition: This is a batch process that generates a report of the discrepancies that exist in the ARMS XREF files between the centralized machine and the distributed computer platforms.

@Purpose: To maintain and keep the centralized and distributed XREFS in synch.

@Notes:

Reports used by the ARMS On-Call staff on daily basis for research and possible manual intervention to fix or resynchronize the centralized and distributed ARMS Cross-Reference database files.

**Process**

Hierarchical numeric ID: 1.1.4.3.5.1

Coded name: CLL749

Name: PGM Consolidate Transaction Data (CLL749)

Comment: @Purpose:

To set up the environment and run the program to generate the cross-reference discrepancy report.

@Operational Method:

- When on the centralized host machine:

1) Check all distributed computer platforms that are ARMS active - defined in system list ACTVARMS, for DDM availability.

If all computer platforms are not available end the program.

2) For each distributed host system that is included in ACTVARMS system list,

copy the cross-reference file to the centralized host machine, where it is consolidated into ZAMXREF.

- When on the DEVELOPMENT machine:

1) For each simulated (dummy DDM) distributed file that is ARMS active - defined in system list ACTVARMS, copy the cross-reference file to the consolidated file, ZAMXREF.

- If not on either of the above mentioned host computer platforms, send program

message 'YOU ARE NOT ON THE DEV OR RARMS MACHINE, PROGRAM CLL749 ENDING'.

**ARMS Process Report**

- Check if there are any records accumulated in ZAMXREF.
  - If there are no records,
    - send message 'THERE WERE NO RECORDS  
TO PROCESS, PROGRAM CLL749 ENDING'
  - Else
    - Override to print to output queue ARMSRPTS
    - CALL program AM0749 to compare the cross references and generate the discrepancy report.
- Return

**Process**

Hierarchical numeric ID: 1.1.4.3.5.9

Coded name:

Name: RPT Exception of Discrepancies Between Centralized Cross-reference and Distributed

Comment: @Purpose:

Report use by the Application Support Staff for the sake of keeping the database in synch.

@Report Distribution:

No automatic distribution. The staff on call has to go to the output queue ARMSRPTS to retrieve the report.

**Process**

Hierarchical numeric ID: 1.1.4.3.5.11

Coded name: AM0749

Name: PGM Print Discrepancies Between Centralized Cross-reference and Distributed (AM0749)

Comment: @Definition:

This is a batch process that generates the ON-CALL report of discrepancies that exist in the ARMS XREF files between the centralized machine and the distributed machines.

@Purpose:

To maintain and keep the centralized and distributed XREFS in synch.

@Operational Method:

- Process the data in the consolidate file against the data in the centralized file. Each record will fall into one of the following categories:
  - 1) Record(s) exist on the distributed system only.
  - 2) Record(s) exist on the centralized system only and no error associated with the transaction.
  - 3) Record(s) exist on both centralized and distributed system, but data not in synch.
- Print the three reports.

**Process**

**ARMS Process Report**

Hierarchical numeric ID: 1.1.4.3.7  
Coded name:  
Name: BAT Added 'I'nsurance Customer  
Comment: @Definition: This activity generates the report listing all the insurance type customers that were added to file CUSTMAST using customer maintenance program AACM07 since the report was last generated.

**Process**

Hierarchical numeric ID: 1.1.4.3.7.1  
Coded name: CLL010  
Name: PGM Generate Report of Added Insurance Customer (CLL010)  
Comment: @Purpose:

To set up the environment and invoke the program to generate a report listing all the insurance type customers that were added to file CUSTMAST using customer maintenance program AACM07 since the report was last generated.

**@Operational Method:**

- Consolidate the data (file CIADDS) from all distributed computer platforms on to the financial application system's host platform and then generate a report from the consolidated data.

**Process**

Hierarchical numeric ID: 1.1.4.3.7.8  
Coded name:  
Name: RPT Added 'I'nsurance Customer  
Comment: @Purpose:

To inform the user of all the insurance type customers that were added in the last week using program AACM07 (ECS Customer Maintenance).

**@Report Distribution:**

Cindy Loomis

**Process**

Hierarchical numeric ID: 1.1.4.3.7.10  
Coded name: IR0100  
Name: PGM Print Weekly Report of Added 'I'nsurance Customer (IR0100)  
Comment: @Purpose:

To generate the report of all the insurance type customers that were added in the last week using program AACM07 (ECS Customer Maintenance).

**@Operational Method:**

1) Read through file LD010 (work file with consolidated data from file CIADDS on each machine) and write one d tail line for each customer in this file to generate the report.



**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.3.12

Coded name:

Name: BAT Missing Open Rental Notification

Comment: @Definition: A batch process that generates a discrepancy report of authorization and associated ticket that are linked yet not completely updated or in synch with the transaction cross-reference file.

@Purpose: To inform the application staff of contracts that are linked to authorizations yet do not contain the authorization information or are not in sync with the transaction cross-reference file.

@When all the pieces are in place the program is capable of generating the necessary request to ARMS to generate the opening rental notification.

**Process**

Hierarchical numeric ID: 1.1.4.3.12.1

Coded name: AML901A

Name: PGM Submit Report Program (AML901A)

Comment: @Purpose:

To submits the report program to generate missing RN report.

@Operational Method:

- Submit the job to call program AMZ001A to detect the missing opening RNS and generate the report for the on call staff to investigate.

@Notes:

- This program is called by a scheduled job (AML901A) in ROBOT at 2100 hrs.

**Process**

Hierarchical numeric ID: 1.1.4.3.12.2

Coded name: AMZ001A

Name: PGM Detect / Generate Missing Open Rental Notification Report (AMZ001A)

Comment: @Definition:

A batch process that generates a discrepancy report of authorization and associated ticket that are linked yet not completely updated or in sync with the transaction cross-reference file.

@Purpose:

To inform the application staff of contracts that are linked to authorizations yet do not contain the authorization information or are not in sync with the transaction cross-reference file.

@Operational Method:

- For every ticket in the file RACMAST that is completely open i.e. CONTRACT DATE has a valid date, STATUS is blank, RMRNO (reservation number) is not blank and TYPE is 'I':

<b>ARMS Process Report</b>
----------------------------

1) Retrieve the reservation master file RACBRMST with the reservation number as the key and compare the ticket numbers in the two files:

CASE 1: If the ticket number in RACBRMST is = 999999, then ignore this ticket and go to the next ticket.

CASE 2: If the ticket number in RACBRMST is not equal to ticket number in RACMAST, then generate message "Ticket linked to Res, but Res points to different Ticket."

CASE 3: If the ticket numbers in the two files match, then:

a) Retrieve supplemental ticket file RACSMAST.

- If no record found,  
then go to next ticket

Else

- If ARMS Flag in RACSMAST (RMARMS)  
and RACBRMST (BRARMS) both are blank,  
then go to the next ticket

Else

- If the flags are different, then report message  
"Ticket and Res linked to  
each other, however ARMS Flags are different"

Else

call cross-reference retrieval program,  
- If call fails, report message  
"cross-reference retrieval failed"  
Else  
use the following decision table.

- C1 : Xref found

C2: Xref not found

C3: Xref status = 'O'

C4: Xref status <> 'O' and <> 'R'

C5: Xref status = 'R'

C6: Transaction exists in AMTRNCTL

C7: Transaction does not exist in AMTRNCTL

If C2; report message "No cross-reference found"

If C1 and C3, go to next ticket

If C1 and C4, report message "Cross-reference status not R"

If C1, C5 and C6, report message "Record exists in file AMTRNCTL"

If C1, C5 and C7; send the opening RN data queue entry and report message "Missing open RN sent to ARMS".

- When the program reports the first message it encounters based on the condition, the next step is to go to the next ticket.

#### Process

Hierarchical numeric ID: 1.1.4.3.12.3

Coded name:

Name: RPT Exception of Missing Open Rental Notification

Comment: @Purpose: Synchronize the ECARS and ARMS database by reporting the inconsistency.

@Report Distribution: Not Defined.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.3.13

Coded name:

Name: BAT Online Reporting Sync Report

Comment: @Definition: A batch process that generates a discrepancy report for the ARMS Online Reporting function. This looks at the data stored in file AM095P which is the primary file for the Online Reporting Screens and compares it with ARMS and ARMS/400 detail files to determine if an out of sync condition exists and needs to be addressed.

@Purose: To inform the ARMS application staff of any out of sync conditions that might cause Trading Partner adjusters to see incorrect information on the Online Reporting screens.

@When: This process is run via ROBOT scheduling in the evening.

**Process**

Hierarchical numeric ID: 1.1.4.3.13.1

Coded name:

Name: PGM CL Wrapper for AMS530A (AML530A)

Comment: @Purpose: This is a CL Wrapper used for the sole purpose of calling AMS530A for creation of the Online Reporting Sync Report.

@Operational Method:

This program does an OVRDBF on file AM097P00 for blocking purposes, adds A4LIB to the library list, OVRPRTF of the report to OUTQ ARMSRPTS, CALLs the program AMS530A and when it returns clears the file AM097P00.

@Files: (CRUD)  
-AM097P00 (-R-D)

**Process**

Hierarchical numeric ID: 1.1.4.3.13.2

Coded name: AMS530A

Name: PGM Online Reporting Sync Report Program (AMS530A)

Comment: @Purpose: This program will look at transactions processed for that day and compare select values between ARMS/400 and ARMS looking for discrepancies. When a difference is found a detail line is written to the report for the use by Oncall personnel to use to be sure that the online report feature stays in sync with ARMS.

@Operational Method:

This program reads through file AM097P00 which is a transactional log created by AM0055V1. This file contains records for both ARMS and ARMS/400 customers and currently the Online Reporting is only used by ARMS/400 customers. Therefore a check is done by reading A4PRF1 to determine if this is a record for an ARMS/400 customer. IF not skip to the next record. If it is an ARMS/400 customer read the associated file record from AM095P00. If a record is found proceed with a sync check. Read the required detail files and those listed in the @Files section.

The following data elements are sync checked:

- Customer Tran. ID
- Branch Claims Office ID
- Claim Number
- Claim Type
- Rental Status
- Adjuster last and first name

<b>ARMS Process Report</b>
----------------------------

Body Shop name  
 Rental start date  
 Rental close date  
 Rental termination date  
 Daily authorized rate  
 Number of days Authorized  
 Dollars covered per Day  
 Policy Maximum Dollars  
 Bill to \$  
 verify correctness of the Surcharge flag  
 IF Invoiced:  
     Total Charges  
     Total Amount Received  
     Billed Amount

Calculated fields checked against AM095P00  
     Number of days in Rental  
     Number of days Behind  
     Number of Extensions received

If any of the above mentioned fields are in an out of sync condition write a detail record to the Sync Report. Continue with the above process until the file AM097P00 is at End of File.

@Notes: Currently the program only checks transactions that are at Open Rental status and is not processing the code for invoiced transactions.

@Files: (CRUD)

- A4XREFL7 (-R--)  
 - A4AUTD (-R--)  
 - A4RPRD (-R--)  
 - AMFDTBL (-R--)  
 - AM095P00 (-R--)  
 - AM097P00 (-R--)  
 - A4SURD (-R--)  
 - A4PRF1 (-R--)  
 - A4IEBT (-R--)  
 - AMTIMEL1 (-R--)  
 - AMTRNLOG (-R--)

@Embedded Data/Constants

Surchg1	=	'Flag is Yes'
Surchg2	=	'No A4SURD rec. found'
Surchg3	=	'Surchg flag'
RntDys1	=	'Days in Rental'
RntDys2	=	'Value in AM095P is not'
RntDys3	=	'correct please verify'
Dysbhnd1	=	'Rental Days Behind'
Dysbhnd2	=	'Value in AM095P is not'
Dysbhnd3	=	'correct please verify'
AdjName	=	'Adjuster Name'
NoXref	=	'No A4XREF for this Tran'

Process

Hierarchical numeric ID: 1.1.4.3.13.3  
 Coded name:

**ARMS Process Report**

Name: RPT Online Sync Report  
Comment:

**Process**

Hierarchical numeric ID: 1.1.4.3.14

Coded name:

Name: BAT Nightly Update Open Tickets for AM095P

Comment: @Definition: A batch process that will update the ARMS Online Reporting database to keep select data elements such as number of days in rental current.

@Purpose: There exists several calculated fields in the database which would become incorrect if a nightly update did not occur. This process maintains the validity and integrity of this data. The file updated is AM095P.

@When: This process is run via ROBOT scheduling in the evening.

**Process**

Hierarchical numeric ID: 1.1.4.3.14.1

Coded name: AML520A

Name: PGM Nightly Update/Write Open Tickets (AML520A)

Comment: @Purpose: To update all open tickets in the ARMS Online Reporting database file to show the correct

number of days in rental and the number of days behind if the extension date has been passed. This will ensure that open rentals that do not have a transaction processed on a given day will be kept up to date.

@Operational Method:

- Retrieve a record from ARMS Online Reporting Detail File AM095P.
- Determine the total number of days in the rental by calculating the difference between today's date and the rental start date.

-IF the termination date is greater than zero, zero out the days behind. Otherwise, the days behind will be equal to the total number of days in the rental less the number of authorized days.

-Update ARMS Online Reporting Detail File AM095P.

@Files: (CRUD)

- AM0095P01 (-RU-)

@Embedded Data/Constants:

**Process**

Hierarchical numeric ID: 1.1.4.4

Coded name:

Name: SA Tools

**ARMS Process Report**

Comment: @Definition: The activity that encompass the manual processes that are performed by various Application Support Staff. These processes range from keeping the environment up and running, troubleshooting transaction problems, to requesting reports.

**Process**

Hierarchical numeric ID: 1.1.4.4.1  
Coded name:  
Name: ONL Generate Application Reports  
Comment: @Definition:

This activity is performed by ARMS Help Desk and ARMS Technical for generating system utilization by Trading Partner and daily billing error report.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.1  
Coded name:  
Name: MNU ARMS Application Reports  
Comment: @Definition: see ONL Generate Application Reports

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17  
Coded name:  
Name: ONL ARMS Reservation Detail by Company Utilization Management Reports  
Comment: @Definition: This activity provides the Application Support Staff the means to produce a utilization management report for a specific Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1  
Coded name:  
Name: ONL Utilization Management Report for Specific Trading Partners BCO (Opt -1)  
Comment: @Definition: Generate a report showing ARMS Trading Partners Utilization of the Automated Rental Management System for specific Branch Claims Office.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1.1  
Coded name: AM0135  
Name: PGM Print Cross - Reference Utilization Report (AM0135)  
Comment: @Purpose:

To print daily utilization report including total reservation, cancelled reservation and rentals, by Trading Partner.

**@Operational Method:**

- Read entire file printing those records that meet the date range supplied to the program.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1.2  
Coded name: AMS002A  
Name: PGM Select Transactions by Trading Partner and BCO (AMS002A)  
Comment: @Purpose:

To select transactions from the cross-reference file, to report for a specified trading partner or branch claims office.

@Operational Method:

- Select those records that meet the date and status criteria.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1.4  
Coded name: CLL003  
Name: PGM Gather / Print Cross - Reference Statistical Report (CLL003)  
Comment: @Purpose:

To generate a statistical transactions report using the cross-reference file.

@Operational Method:

- Execute the process that selects those records that meet the criteria for the utilization report, program AMS002A
- Override printer file to outq ARMSRPTS.
- Execute the process that print the utilization report, program AM0135.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1.5  
Coded name: AM0136  
Name: PGM Display / Select Available Trading Partners Claims Office (AM0136)  
Comment: @Purpose:

To display a list of available claims offices from which the Application Support Staff can select from.

@Operational Method:

- Build a file / list of available Trading Partners.
- When requested, display the filter allowing the user to select specific Trading Partners Branch Claims Office.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.17.1.6  
Coded name: CLL001  
Name: PGM Gather / Print ARMS Reservation Detail Utilization Management Reports (CLL001)

**ARMS Process Report****Comment:**      **@Purpose:**

To allow the "ARMS Technical" staff to monitor trading partner's usage of the ARMS system via a cross-reference report of authorizations approved through ARMS. The filter criterion can specify the trading partner profile (all BCO's, Branch Claims Offices) or particular BCO's within a profile

**@Operational Method:**

- Print a detailed report by Rental Management Trading Partner company of the current statistics of approved authorizations via ARMS.

**@Files:**                      (CRUD)

- AMXREF                      (-R--)  
- ARMSPR2                    (-R--)  
- AMCLSTBL                  (-R--)  
- AMAUTD                    (-R--)  
- OFFDGRPPF                (-R--)  
- AMMNTLOG                  (-R--)

**@Note:** Used by ARMS Technical Manager to see increase or decrease in ARMS usage volume by each Rental Management Trading Partner.

**Process****Hierarchical numeric ID:**      1.1.4.4.1.17.2**Coded name:****Name:** ONL Utilization Management Report for Trading Partner (Opt - 2)**Comment:**      **@Definition:** Generate a report showing ARMS Trading Partners Utilization of the Automated Rental Management System.**Process****Hierarchical numeric ID:**      1.1.4.4.1.17.2.1**Coded name:** AMS003A**Name:** PGM Select Transactions for a Trading Partner (AMS003A)**Comment:**      **@Purpose:**

To select transactions from the cross-reference file based on certain criteria specified via a job file AML136JA.

**@Operational Method:**

- Select those records that meet the date and status criteria.

**Process****Hierarchical numeric ID:**      1.1.4.4.1.17.2.3**Coded name:** CLL001A**Name:** PGM Gather / Print Cross - Reference Statistical Report (CLL001A)**Comment:**      **@Purpose:**

To set the environment and execute the programs that select transactions from the cross-reference file, and print the actual report.

**@Operational Method:**



**ARMS Process Report**

- Execute the program to select the transactions specified by the user.
- Override the print file for the report generator
- Execute the program to print the utilization report.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.18  
Coded name:  
Name: ONL Generate Report of Errors Sent by a Trading Partner on an Electronic Invoice  
Comment: @Purpose: A process where the Application Support Staff can generate a list of all the sent electronic invoices that were rejected by the trading partner.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.18.1  
Coded name: CLL015  
Name: PGM Generate Daily Billing Error Report (CLL015)  
Comment: @Purpose:

To control the generation of the daily billing error report.

**@Operational Method:**

- Execute the program to build the work file with the billing error information
- Execute the program to print the billing error report.

**Process**

Hierarchical numeric ID: 1.1.4.4.1.18.2  
Coded name:  
Name: PGM ARMS System Control (AAMAIN)  
Comment: @Definition: A single source driver for the entire ARMS System Control 'menu'. Allowing the Application Support Staff to maintain the ARMS environment; Trading Partner profiles; application tables;

**Process**

Hierarchical numeric ID: 1.1.4.4.1.18.6  
Coded name: AM5010  
Name: PGM Construct Billing Error File (AM5010)  
Comment: @Purpose:

To build a work file, selecting records according to the specified date range of error records received from the Trading Partners for sent invoices.

**@Operational Method:**

- Read entire received bill error file selecting those records that match the date range criteria.
- When a match is found retrieve the associated sent transaction.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.4.1.18.7  
Coded name: AM5015  
Name: PGM Print Billing Error report (AM5015)  
Comment: @Purpose:

To print a list of errors including comments received from the Trading Partner that fall in the specified date range.

@Operational Method:

- Process the work file, print the transaction that rejected and why.

**Process**

Hierarchical numeric ID: 1.1.4.4.2  
Coded name:  
Name: SCR Work with Robot Scheduler (RBM)  
Comment: @Definition:

This activity routes the ARMS on-call staff to the job scheduler main menu where inquiry about specific jobs can be made.

**Process**

Hierarchical numeric ID: 1.1.4.4.3  
Coded name:  
Name: ONL Work with Transaction Exception  
Comment: @Definition:

This activity is performed by the ARMS on-call staff to be both reactive and proactive to data base discrepancies between the centralized ARMS system and distributed system as well as ARMS system and Rental system and ARMS/400.

**Process**

Hierarchical numeric ID: 1.1.4.4.3.1  
Coded name:  
Name: SCR Work with Job (LAMZ001A)  
Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the missing rental notification report.

@Notes:

- This report is presently the proactive approach to the missing rental notification problem. The problem is the result of deficiencies in handling certain scenarios;

- Report is generated on the AEFIKLMNPSTWY host computer platforms.

**Process**

Hierarchical numeric ID: 1.1.4.4.3.2  
Coded name:  
Name: SCR Work with Job (AML900)

**ARMS Process Report**

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the discrepancy report between ARMS and Rental.

## @Notes:

- This program runs a series of queries that compares the data found in the Ticket file to ARMS cross-reference, as well as the data found in the reservation file to ARMS cross-reference.

- The report based on the reservation file contains many of transaction that should not be include. This is the result of using the open ticket date on the reservation file instead of the contract date from the ticket.

- The report does show when two reservation points to only one ticket.

- Report is generated on the AEFIKLMNPSTWY host computer platforms.

## Process

Hierarchical numeric ID: 1.1.4.4.3.3

Coded name:

Name: SCR Work with Job (L749)

Comment: @Definition: Provides the on-call staff an option for retrieving the ARMS cross-reference discrepancy report.

## @Notes:

- Report is generated on RARMS machine.

## Process

Hierarchical numeric ID: 1.1.4.4.3.4

Coded name:

Name: SCR Search for Spool Entries (DSPOLE)

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the electronic billing error report that has been archived

## @Notes:

- Report is generated on AEFIKLMNPSTWY host computer platforms.

## Process

Hierarchical numeric ID: 1.1.4.4.3.5

Coded name:

Name: SCR Work with All Spooled Files (A4ERROR)

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the discrepancy report between ARMS and ARMS/400.

## @Notes:

- Report runs on RARMS only.

- Libraries A4LIB and ELPGMR required.

## Process

Hierarchical numeric ID: 1.1.4.4.3.6

Coded name:

Name: SCR Work with All Spooled Files (A4XREF)

**ARMS Process Report**

Comment: @Definition: Provides the ARMS on-call staff an option for retrieving the report of ARMS/400 transactions at a sent status.

## @Notes:

- Reactive approach for handling errors.
- Report runs on RARMS only.
- Libraries A4LIB and ELPGMR required

**Process**

Hierarchical numeric ID: 1.1.4.4.3.7

Coded name:

Name: MNU ONCALL Reports

Comment: @Definition: see ONL Work with Transaction Exception

**Process**

Hierarchical numeric ID: 1.1.4.4.4

Coded name:

Name: SCR Access ARMS/400 Software (A4000RMS)

Comment: @Definition: This activity allows the ARMS on-call staff to access the ARMS/400 software for researching problems.

@Notes: This access path to the ARMS/400 interactive system allows a user to see the screens and information in the same way that the adjuster or in-house Enterprise employee would. Further this was created initially for use by the Rental Management Services team headed by Jim Hillerman.

**Process**

Hierarchical numeric ID: 1.1.4.4.5

Coded name:

Name: ONL Perform ARMS Environment Maintenance

Comment: @Definition:

This activity is performed by the ARMS on-call staff for managing the application environment (ARMS subsystem).

**Process**

Hierarchical numeric ID: 1.1.4.4.5.1

Coded name:

Name: ONL Start Individual Trading Partner Communication Programs

Comment: @Definition:

This activity is performed by the staff on call to select and re-start the communication programs for a given Trading Partner when they have abnormally ended.

## @Notes:

- Runs on RARMS

**Process**

**ARMS Process Report**

Hierarchical numeric ID: 1.1.4.4.5.1.1  
Coded name: CCARMS  
Name: PGM Select Company / Machine (CCARMS)  
Comment: @Purpose:

To allow to select and return the selected Trading Partner send and/or receive detail to provide a way to re-start the communication programs for a given Trading Partner when they have abnormally ended.

**@Operational Method:**

- Display the list of computer platforms where ARMS can be started.
- For each machine selected, read the ARMS profile attributes for that trading partner and display the send and receive and associated translator and mapper program START FLAGS for the user to select which programs to start.
- Return the flag values as selected by the user.

**Process**

Hierarchical numeric ID: 1.1.4.4.5.1.2  
Coded name: AMSTARTR  
Name: PGM Start Specific Trading Partner Send & Receive pgms (AMSTARTR)  
Comment: @Purpose:

To submit jobs to start the send and receive and associated programs for each Specified Trading Partner without a VAN in the profile file (ARMSPR3).

**@Operational Method:**

- If the company id is passed,
  - retrieve the communication program technical detail for the passed company id
  - Else all companies programs to be started
  - read the file ARMSPR3 (ARMS Technical Profile) sequentially and perform the following steps for each qualified company
- Based on the profile, the passed flags and some hard coded logic, submit jobs to start appropriate communication programs at appropriate scheduled times.

**Process**

Hierarchical numeric ID: 1.1.4.4.5.2  
Coded name:  
Name: ONL Start Transaction Distribution Programs for Specific(s) Machine  
Comment: @Definition: A tool which provides the Application Support Staff the means necessary to start the distribution program from the centralized host machine to the distributed host machine.

**@Notes:**

- Runs on RARMS

**Process**

**ARMS Process Report**

Hierarchical numeric ID: 1.1.4.4.5.2.2  
Coded name: DQAAARMS  
Name: DTQ Selected Computer platforms with Start Flags  
Comment: @Definition:

This data queue is created by program AAMAIN in library QTEMP to retrieve the selected computer platforms and associated start flags. The computer platforms are selected in program CCARMS where the entries are sent to this data queue.

Essentially this data queue acts as a data area to communicate between the two programs AAMAIN and CCARMS.

**Process**

Hierarchical numeric ID: 1.1.4.4.5.3  
Coded name:  
Name: ONL Start the ARMS to Rental Systems Interfaces  
Comment: @Definition: A tool which provides the Application Support Staff the means necessary to start all of the ARMS to Rental Systems Interfaces for a single distributed host machine.

**@Note:**

- Runs on A,C,E,F,I,K,L,M,N,P,S,T,W,Y

**Process**

Hierarchical numeric ID: 1.1.4.4.5.3.1  
Coded name: AMSTARTE  
Name: PGM Start the ARMS - Rental System Interfaces (AMSTARTE)  
Comment: @Purpose:

To allow the user to start the rental system interface programs for a particular machine.

**@Operational Method:**

... - Submit jobs to start each of the ECARS interface programs.

@Notes: Interface include: EC/CC00ATVn, EC/CC00CNVn, EC/CC00EXVn, EC/CC00ACVn, EC/CC00RAVn, EC/CC00RCVn, EC/CC00REVn, EC/CC00RNVn, EC/CC00TRVn, EC00CMVn, EC00VMVn, CC00INVn, AM0061Vn, AM0060Vn

**Process**

Hierarchical numeric ID: 1.1.4.4.5.4  
Coded name: WRKCFGSTS  
Name: ONL Work with a Specific Trading Partner Communication Line (WRKCFGSTS)  
Comment: @Definition: A command that allows the Application Support Staff to display and work with the status of a Specific Trading Partner's communication line.

**@Notes:**

- Run on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.5.5

**ARMS Process Report**

Coded name: WRKLIND

Name: ONL Work with a Specific Trading Partner Communication Line Description (WRKLIND)

Comment: @Definition: A command that allows the Application Support Staff to display and work with the line description of a Specific Trading Partner.

@Notes:

- Run on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.5.6

Coded name: WRKHDWRSC

Name: ONL Work with Local Communication Resources (WRKHDWRSC)

Comment: @Definition: A command that shows information about local communications resources on the system and the associated configuration objects, including all communication IOPs, IOAs, and ports installed on the system.

@Notes:

- Run on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.5.7

Coded name:

Name: ONL Work with the ARMS Subsystem (WRKSBSJOB)

Comment: @Definition: A command that shows all jobs running in the Application Specific subsystem (ARMS).

@Note: Subsystem is the same on all computer platforms. Jobs running in the subsystem differs from centralized host computer platforms to distributed host computer platforms to the Financial Application systems host computer.

- Runs everywhere

**Process**

Hierarchical numeric ID: 1.1.4.4.5.8

Coded name: NEWARMS

Name: ONL Start new Subsystem Program (NEWARMS)

Comment: @Purpose: A command that allows the Application Support Staff to start a user specified, non-data required program.

@Notes:

- Runs everywhere

**Process**

Hierarchical numeric ID: 1.1.4.4.5.9

Coded name:

Name: ONL Shutdown ARMS Environment for a Specific Computer platforms

Comment: @Purpose: The process by which the ARMS Application Support Staff ends the processes running on a specific distributed computer platforms.

@Notes: This usually is the result of a computer platforms that is experiencing hardware difficulties and OPS is wants to bring it down.

ARMS Process Report
---------------------

- Run on RARMS only

**Process**

Hierarchical numeric ID: 1.1.4.4.5.9.1

Coded name: AMSHUTDWN1

Name: PGM Send Shutdown Request (AMSHUTDWN1)

Comment: @Purpose: The process where a shutdown request is sent by the ARMS on-call staff, for a specific distributed computer platforms, to end the running ARMS environment.

@Operational Method:

- If running on the RARMS computer platforms, send data entry to DQAMAPI with the specified computer platforms prefix as the key.

@Notes:

- \*DOWN000000000000SD\*DOWN1 = Shutdown data queue entry.

**Process**

Hierarchical numeric ID: 1.1.4.4.5.10

Coded name:

Name: ONL Display the Number of Shutdown the Network Requires

Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of shutdown records the ARMS network requires before shutting down.

@Notes:

- The number of shutdown records is derived from the number of distributed computer platforms it is running on. Presently, the number is 14.

- DSPDTAARA AM001A

- Runs on RARMS only

**Process**

Hierarchical numeric ID: 1.1.4.4.5.11

Coded name:

Name: ONL Display the Number of Shutdown Network Received

Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of shutdown records the ARMS network has received.

@Notes:

-If not enough shutdown records have been received the on-call staff can generate the missing records to cause the ARMS network to shutdown.

\*\*The most common cause for insufficient shutdown records happens when one of the distributed computer platforms is brought down at an unscheduled time.

- DSPDTAARA AM002A
- Runs on RARMS

**Process**



<b>ARMS Process Report</b>
----------------------------

Hierarchical numeric ID: 1.1.4.4.5.12  
Coded name:  
Name: ONL Work with the ARMS Job Queue (WRKJOBQ)  
Comment: @Definition: Process by which the ARMS Application Support Staff can monitor the ARMS job queue.

@Notes:  
- Run everywhere

**Process**

Hierarchical numeric ID: 1.1.4.4.5.13  
Coded name:  
Name: ONL Display the Number of any ARMS-connected VAN Receivers Started  
Comment: @Definition: Process by which the ARMS Application Support Staff can see the number of ARMS-connected VAN communication receiver jobs started.

@Notes:  
- This count is usually 1. However, if a second receiver is started the data queue and file that the received transmissions are placed into is AMVN101T instead of AMVN1011.

- DSPDTAARA AM003A  
- Run on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.5.14  
Coded name:  
Name: ONL Display the Number of Transaction Translators Running  
@Definition: The process by which the ARMS on-call staff can see the number of translators running.

@Notes:  
- This is important when shutting down the ARMS network normally. If the value is not equal to zero then a translator is active and the ARMS network will NOT shutdown.

\*\* If the translator was ended abnormally the count is not adjusted.

- DSPDTAARA DASDV1  
- Run on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.5.15  
Coded name:  
Name: MNU Work with ARMS Environment  
Comment: @Definition: see ONL Perform ARMS Environment Maintenance

**Process**

Hierarchical numeric ID: 1.1.4.4.6  
Coded name:  
Name: ONL Perform ARMS/400 Environment Maintenance

**ARMS Process Report**

Comment: @Definition:

This activity is performed by the ARMS on-call staff for managing the ARMS/400 application environment.

@Notes:

- Runs on RARMS only

**Process**

Hierarchical numeric ID: 1.1.4.4.6.1

Coded name:

Name: ONL Work with the ARMS/400 Subsystem (WRKSBSJOB)

Comment: @Definition: A command that shows all jobs running in the Application Specific subsystem (A4BAT).

@Note: Subsystem only on RARMS

**Process**

Hierarchical numeric ID: 1.1.4.4.6.2

Coded name:

Name: ONL Work with the ARMS/400 Job Queue (WRKJOBQ)

Comment: @Definition: Process by which the ARMS Application Support Staff can monitor the A4BAT job queue.

@Notes:

- Run RARMS only

**Process**

Hierarchical numeric ID: 1.1.4.4.6.3

Coded name: NEWARMS400

Name: ONL Start new Subsystem Program (NEWARMS400)

Comment: @Purpose: A command that allows the Application Support Staff to start a user specified, non-data required program.

@Notes:

- Runs RARMS only

**Process**

Hierarchical numeric ID: 1.1.4.4.6.4

Coded name:

Name: MNU Work with ARMS/400 Environment

Comment: @Definition: see ONL Perform ARMS/400 Environment Maintenance

**Process**

Hierarchical numeric ID: 1.1.4.4.7

Coded name: AAMAIN

Name: MNU ARMS Main Menu (AAMAIN)

Comment: @Definition: A consolidated menu where by the ARMS on-call staff can find and access support tools needed to perform their daily activities.

**ARMS Process Report****@Notes:**

- Rental System Help Desk and ARMS Application Support Service use the menu.
- Tools are group into four main categories, with the further group into subcategories and so on based on their related functions:
    - Reports
      - Application
      - On-call
    - Applications
      - Robot Scheduler
      - ARMS/400
    - Maintenance / Inquiry
      - ARMS Environment
        - Startup Controls
        - Shutdown Controls
        - Communication Status
        - Other
      - ARMS/400 Environment
      - Transaction Log
      - Data Queue
      - Reservation/Ticket Information
      - Purge Progress
      - File
    - Other

**Process**

Hierarchical numeric ID: 1.1.4.4.8

Coded name:

Name: ONL Inquire on Transaction Log

Comment: @Definition:

This activity is used by on-call and help desk when troubleshooting discrepancies between the Trading Partner and Rental Systems

**Process**

Hierarchical numeric ID: 1.1.4.4.8.1

Coded name:

Name: MNU ARMS System Control Transaction Log

Comment: @Definition: see ONL Inquire on Transaction Log

**Process**

Hierarchical numeric ID: 1.1.4.4.8.11

Coded name:

Name: ONL Inquiry of Transactions Received from a Trading Partner

Comment: @Definition: The process which allows the Application Support Staff to select specific transactions received from the Trading Partner's for inquiry.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.11.1

Coded name: CCAM12

Name: PGM Transaction Receive Log Inquiry (CCAM12)

**ARMS Process Report**

Comment: @Purpose:

To allow the Application Support Staff to select specific transactions received from the Trading Partners for inquiry.

@Operational Method:

- Display a list of all the transmission received from specific Trading Partners.

- Upon the select of an individual transmission, display the control/data records the make up that transmission.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.12

Coded name:

Name: ONL Inquiry of Error Transactions Sent by Trading Partner

Comment: @Definition: A process where the Application Support Staff has the ability to view error transactions sent by Trading Partners.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.12.1

Coded name: CCAM13

Name: PGM Receive / Billing Error Inquiry (CCAM13)

Comment: @Purpose:

To display either the billing errors received from the Trading Partners or the non - billing errors. This is helpful for troubleshooting.

@Operational Method:

- Based on the parameter passed to the program open either the received error log file or the billing error log file.

.. - Read all records from the file, displaying the records on screen.

- Upon the selection of an individual transmission, display the error code received.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.13

Coded name:

Name: ONL Inquiry of Transaction Sent by the Trading Partner but Erred / Rejected

Comment: @Definition: The process where the Application Support Staff can display the errors that have been sent to the Trading Partners.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.13.1

Coded name: CCAM14

Name: PGM Error Log Inquiry (CCAM14)

**ARMS Process Report**

Comment: @Definition: The process where the Application Support Staff can display the errors that have been sent to the Trading Partners.

@Purpose: Used in trouble shooting problems

@Operational Method:

- Display a list of the errors received by Enterprise from a specific Trading Partner.

- When a specific transmission is selected, display the data records associated with that transmission.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.14

Coded name:

Name: ONL Inquiry / Resend of Transaction(s) ARMS Sent to Trading Partner

Comment: @Definition: A process where by the Application Support Staff can display transaction sent to the Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.14.1

Coded name: CCAM15

Name: PGM Send Log Inquiry and Resend (CCAM15)

Comment: @Definition: A program where by the Application Support Staff can display transaction sent to the Trading Partner. It may, if needed, select a sent transaction, alter the data, and resend the transaction to the Trading Partner for processing.

@Purpose: This is useful when troubleshooting a problem

@Operational Method:

- Read the sent transaction log file, display a list on screen for the user.

- Upon the selection of one of the sent transmissions, display the data/control record that make up that single transmission.

- Once a transmission has been selected, the user has the option of changing the data in the transmission and resending it.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.15

Coded name:

Name: ONL Audit of Transaction Activity

Comment: @Definition: A process where by the Application Support Staff is able to display the electronic history of a specific transaction.

@Purpose: Very useful in problem determination.

**Process**

**ARMS Process Report**

Hierarchical numeric ID: 1.1.4.4.8.15.1  
Coded name: CCAM16  
Name: PGM Timeline Inquiry (CCAM16)  
Comment: @Definition: A program that will display the electronic history of a specified transaction.

@Purpose: Very useful in problem determination.

@Operational Method:

- Retrieve the transaction information via 1 of 4 ways:
  - Vendor Transaction Id
  - Company & Customer Transaction ID
  - Location & Reservation
  - Location & Ticket
- Retrieve all sent and received transaction with the same vendor transaction id and display this information.
- When the user selects an individual transmission, display the details of that transmission.
- When F2 is pressed, a printout is generated with all the information associated to the transaction.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.15.2  
Coded name: CLL800  
Name: PGM Cross-reference Maintenance Log Driver (CLL800)  
Comment: @Purpose: The driver program that sets up the print file before executing the program to print the cross-reference maintenance log.

@Operational Method:

- Override print file
- Execute program that generates the maintenance log report

**Process**

Hierarchical numeric ID: 1.1.4.4.8.15.3  
Coded name: AM0800R  
Name: PGM Print Database / Cross-Reference Maintenance Log (AM0800R)  
Comment: @Purpose: To take a 'snapshot' of the existing database for a single transaction.

@Operational Method:

- Retrieve all associated database records for the single vendor transaction id and print the information.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.16  
Coded name:  
Name: ONL Inquiry / Resend of an Electronic Invoice

**ARMS Process Report**

Comment: @Description: A process where by the Application Support Staff can correct an electronic invoices and send it to the Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.4.4.8.16.1

Coded name: AAAM15

Name: PGM Billing Log Inquiry and Resend (AAAM15)

Comment: @Purpose: To select/display records for billing corrections and to RESEND to the customer.

@Operational Method:

- The user must first select the order in which the data will be displayed (GPBR, TICKET# or LOCATION, TICKET#).
- Then the user will select the item to be changed and make the change. The program then recalculates the totals for the trailer record and redispays all. The user can then choose to resend this billing to the customer by pressing CMD-9 twice.
- This will write the header, all corrected detail, and the trailer records to AMPACK. Also will write the APPD01 record before the header record.

**Process**

Hierarchical numeric ID: 1.1.4.4.9

Coded name:

Name: ONL Perform Data Queue Maintenance

Comment: @Definition:

This activity is performed by the ARMS on-call staff to support the application by manipulating the data (transactions) in the environment.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.1

Coded name: ARMSONCL

Name: SCR ARMS Data Queue Monitor (ARMSONCL)

Comment: @Definition: A process from which the ARMS on-call staff can monitor the number of messages waiting to process and where.

@Notes:

- Run everywhere

**Process**

Hierarchical numeric ID: 1.1.4.4.9.2

Coded name: A4ONCL

Name: SCR ARMS/400 Data Queue Monitor (A4ONCL)

Comment: @Definition: A process from which the ARMS on-call staff can monitor the number of messages waiting to process and where.

@Notes:

- Run RARMS only

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.4.9.3

Coded name: ANDQLDDQ

Name: ONL Load Data Queue with Message from Data Queue (ANDQLDDQ)

Comment: @Definition: A process where the ARMS on-call staff can read and load data queue message from and to a specified data queue.

@Purpose: Provide the application development staff a way to perform load balancing when problems with external communications has resulting if a backlog of un-sent transactions.

@Note:

- User specifies the number of messages to read, starting at the first entry.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.4

Coded name: ANDQLDPF

Name: ONL Load Data Queue with Message from a Physical File (ANDQLDPF)

Comment: @Definition: A process where the ARMS on-call staff can read record from a specified physical file and load data queue message to a specified data queue.

@Purpose: Provide the application development staff a way to generate the corresponding data queue entry for the data records.

@Notes:

**Process**

Hierarchical numeric ID: 1.1.4.4.9.5

Coded name:

Name: ONL Retrieve Data Queue Information

Comment: @Definition: A process that allows the ARMS on-call the option to retrieve information about the data queue, particularly the number of messages.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.6

Coded name: RGZDTAQ

Name: ONL Reorganize Data Queue (RGZDTAQ)

Comment: @Definition: The process where by the ARMS on-call staff can delete and recreate a data queue without destroying the data queue entries that might still exist.

@Purpose: Data queue processing some known traits - If the number of messages grow to more than 100, the size of the data queue expands. The only way to reclaim the space is to delete and recreate the data queue. Its a performance issue.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.7

Coded name:

Name: MNU ARMS System Data Queue Maintenance



**ARMS Process Report**

Comment: @Definition: see ONL Perform Data Queue Maintenance

**Process**

Hierarchical numeric ID: 1.1.4.4.9.8

Coded name:

Name: ONL Work with Application Data Queues

Comment: @Definition: A process that provides the Application Support Staff easy access to manage / manipulate the data being passed between processes.

@Purpose: Provide easy access of the APIs to display, dump, and clear a data queue as well as add a data queue entry through the use of a display screen.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.8.1

Coded name: ANDQMNEU

Name: MNU Application Development Data Queue Menu

Comment: @Definition: A menu that provides the Application Support Staff easy access to manage/manipulate the data being passed between processes.

@Purpose: Provide easy access of the APIs to display, dump, and clear a data queue as well as add a data queue entry through the use of a display screen.

**Process**

Hierarchical numeric ID: 1.1.4.4.9.8.3

Coded name:

Name: SCR Add a Data Queue Entry

Comment: @Definition: A tool that allows the Application Support Staff to enter data queue entries as needed.

**Process**

Hierarchical numeric ID: 1.1.4.4.10

Coded name:

Name: ONL Perform Application File Maintenance

Comment: @Definition:

This activity is performed by ARMS Technical to maintain various application table files (i.e. BCO, Trading Partner Profile)

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1

Coded name:

Name: ONL Maintain Trading Partner Profile Global Attributes

Comment: @Definition: The ARMS Technical Staff uses this activity to maintain the ARMS customer global profile information like how the customer connects to Enterprise, what types of transactions the customer receives and/or sends, what is the time window for the customer during which batch invoices can be sent etc.

---

**ARMS Process Report**

---

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.1  
Coded name: AM0700  
Name: PGM Maintain ARMS Trading Partner Profiles (AM0700)  
Comment: @Purpose:

To allow the user to maintain all the company profile attributes related to transactions and communications.

**@Operational Method:**

This program displays a list of all the companies that have been set up in the ARMS system and has options for the user to add, change, copy, delete or display profiles.

There is a separate function key for each classification of the profile attributes and a separate program maintains each classification. The following functions are available:

F7=Appl Specific  
F8=Name Addr  
F9=Tech Data  
F10=File/Field Ctrl  
F14=ECARS Data  
F16=Claims  
F17=Discount  
F18=Underage Driver

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.2  
Coded name: AM0701  
Name: PGM Maintain ARMS Trading Partner Profile Application Specific Information (AM0701)  
Comment: @Purpose:

To maintain the ARMS Application Specific Data that defines how the trading partner host system interacts with the ARMS system. This file is essentially contains an electronic agreement between the two.

**@Operational Method:**

- This program gets invoked when the user presses function key F7 to maintain the application specific data from the main screen displayed by controlling program AM0700.

- The user can change all the profilable attributes for any company. The attributes and their description are retrieved from the ARMS field table file AMFDTBL.

**@Files:**

ARMSPR1 (-RU-)  
ARMSPR2 (-R--)  
AMFDTBL (-R--)

**@Notes:**

**ARMS Process Report**

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.3

Coded name: AM0702

Name: PGM Maintain ARMS Trading Partner Profile Contact Information (AM0702)

Comment: @Purpose:

To maintain the ARMS Customer Address Data for the insurance company.

@Operational Method:

- This program gets invoked when the user presses function key F8 to maintain the company name and address information from the controlling program AM0700 main display.

- The user can change the customer's name, address, phone and fax number of the insurance company's corporate office.

- The file ARMSPR2 is keyed by company profile and customer code which allows for more than one record for every profile.

@Notes:

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.4

Coded name: AM0703

Name: PGM Maintain ARMS Trading Partner Profile Technical Information (AM0703)

Comment: @Purpose:

To maintain the ARMS technical connection information for connecting Enterprise ARMS hostsystem to trading partner host system.

@Operational Method:

- This program gets invoked when the user presses function key F9 to maintain the company technical information from the controlling program AM0700 main display.

- The user can change the customer's connection specific information.

@Notes:

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.5

Coded name: AM0704

Name: PGM Maintain ARMS Trading Partner Profile File/Field Controls (AM0704)

Comment: @Purpose:

**ARMS Process Report**

To maintain the insurance company's field control table that determines which fields are mandatory to send and or receive.

**@Operational Method:**

- This program gets invoked when the user presses function key F10 to maintain the field control information from the controlling program AM0700 main display.

- The user can change the customer's field control table, ARMSPR4.

**@Notes:**

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.6

Coded name: AM0705

Name: PGM Maintain ARMS Trading Partner Profile ECARS-Specific Information (AM0705)

Comment: @Purpose:

To maintain the insurance company's ECARS application system specific data like which billing program to call when closing a ticket.

**@Operational Method:**

- This program gets invoked when the user presses function key F14 to maintain the ECARS information from the controlling program AM0700 main display.

- The user can change how ECARS system interacts with the customer company's authorized rentals.

**@Notes:**

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.7

Coded name: AM0706

Name: PGM Maintain ARMS Trading Partner Profile Claims Connection Specific Information (AM0706)

Comment: @Purpose:

To maintain the insurance company's Claims Connection application system specific data like which billing program to call when closing a ticket.

**@Operational Method:**

- This program gets invoked when the user presses function key F14 to maintain the Claims information from the controlling program AM0700 main display.

- The user can change how Claims Connection system interacts with the customer company's authorized rentals.

**@Notes:**

**ARMS Process Report**

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.8  
Coded name: AM0707  
Name: PGM Maintain ARMS Trading Partner Profile Volume Discount Levels (AM0707)  
Comment: @Purpose:

To maintain the insurance company's discounts based on transaction volumes.

**@Operational Method:**

- This program gets invoked when the user presses function key F17 to maintain the Discount information from the controlling program AM0700 main display.
- The user can change how much discount a company gets at what threshold levels of transactions.

**@Notes:**

The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.1.9  
Coded name: AM0708  
Name: PGM Maintain ARMS Trading Partner Profile Underage Control by State/Province (AM0708)  
Comment: @Purpose:

To maintain the underage surcharge amount and age that is considered underage for each profile and state combination.

**@Operational Method:**

- This program gets invoked when the user presses function key F18 to maintain the Underage information from the controlling program AM0700 main display.
- The user can change the amount to charge for underage drivers and the legal age for that state by company profile.

**@Notes:**

- This file is currently empty and no company has been so customized.
- The changes are then made online to the distributed files via DDM links.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.3  
Coded name:  
Name: ONL Maintain Application Transaction Error Code Table  
Comment: @Definition: A process by which the application support staff can maintain ARMS transaction error code table

<b>ARMS Process Report</b>
----------------------------

**Process**

Hierarchical numeric ID: 1.1.4.4.10.3.1  
Coded name: AAEC01  
Name: PGM Maintain Error Code Table (AAEC01)  
Comment: @Purpose: Allow the application support staff to maintain the transaction error code table.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.4  
Coded name:  
Name: ONL Maintain Car Class Cross-Reference  
Comment: @Definition: A process whereby the application support staff maintains the car class (make/model) cross-reference for every Trading Partner.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.5  
Coded name:  
Name: ONL Maintain Trading Partner Profile Geographic/Regional Attributes  
Comment: @Definition: This activity is used by the ARMS Technical Staff members to maintain the cross-reference between the ARMS customer branch claims office (BCO) and Enterprise customer assigned to the BCO.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.5.1  
Coded name: AAAM10  
Name: PGM Maintain Trading Partner Profile Individual Office Locations Attributes (AAAM10)  
Comment: @Purpose:

To maintain the cross-reference table between the insurance customer's branch claims offices and Enterprise customer numbers assigned to those branch claims offices.

**@Operational Method:**

This program reads file AMXBCO and displays the company id and branch claims offices in that order in a subfile display along with the assigned Enterprise customer numbers and claims office description.

The user can position to a particular profile and then choose to maintain details for or delete a particular company id-claims office pair.

When the user selects to maintain, another screen displays the following attributes for the pair:

ARMS Profile ID . . . . . :  
Customer Branch Claims Office . :  
Enterprise Customer Number. . . :  
BCO Description . . . . . :

ARMS Process Report
---------------------

Country Code . . . . . :  
 Electronic & Paper Bills. . . . : (Yes or No)  
 Include Airport Locations.....  
 ARMS Special Condition . . . . . :  
 Default Node ID..... :  
  
 ARMS Active Control . . . . . : Effective start and end dates  
 ARMS Billing Active Control . . . : - " -  
 ARMS Payments Active Control. . . : - " -

The user has the ability to delete a pair by selecting a 'D' against the pair or another pair can be added by function key F6.

## Process

Hierarchical numeric ID: 1.1.4.4.10.6  
 Coded name:  
 Name: ONL Maintain Application Master Files/Tables  
 Comment: @Definition: This activity allows the user to maintain the master files that may contain customized data for a certain company or certain default information to use when generating transactions.

## Process

Hierarchical numeric ID: 1.1.4.4.10.6.1  
 Coded name: AMS004A  
 Name: PGM Maintain Surcharge Table (AMS004A)  
 Comment: @Purpose:

To allow the user to set up automatic surcharge proprietary formats (SURD01) by company profile and state. These are surcharges that the company authorizes Enterprise to automatically charge against a rental authorized by that company.

## @Operational Method:

The program displays the surcharge format detail fields for a profile in the header section and all the states in the detail section using a standard subfile approach and allows the user to add, change, copy or delete records.

## Process

Hierarchical numeric ID: 1.1.4.4.10.7  
 Coded name:  
 Name: MNU Maintain Application Files  
 Comment: @Definition: see ONL Perform Application File Maintenance

## Process

Hierarchical numeric ID: 1.1.4.4.10.9  
 Coded name:  
 Name: ONL ARMS/400 Training Administration  
 Comment: @Definition: Process by which the Application Support Staff manage the ARMS/400 Training System.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.4.10.9.1  
Coded name: A4TRNGMN  
Name: MNU ARMS/400 Training Administration Menu (A4TRNGMN)  
Comment: @Definition: A process which enables the Application Support Staff to manage the training environment / data. From setting up claims office and testing scenarios to checking the status of the system.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.10  
Coded name:  
Name: ONL Maintain Vehicle Class / Rate Table  
Comment: @Definition: Provide the means for the Application Support Staff to maintain Trading Partner's rate to Enterprise's car classes.

**Process**

Hierarchical numeric ID: 1.1.4.4.10.10.1  
Coded name: CCAM11  
Name: PGM Vehicle Class / Rate Table Maintenance (CCAM11)  
Comment: @Purpose:

To provide the means for the Application Support Staff to maintain Trading Partner's rate to Enterprise's car classes cross-reference table (including ARMS/400 Customers).  
@Operational Method:

- Display the details in file AMCLSTBL or A4CLSTBL and allow the user to maintain the data and synchronize the data on all computer platforms via DDM links.

The first screen displays the Company ID, ARMS class, Enterprise class, Class description and the X12 car class.

... Options available from this screen:

1. Position to a particular Company ID
2. Add a new ARMS class to an existing Company ID or a new Company ID, in which the Company will be created.
3. Change a current ARMS class for a Company ID.
4. Delete a current ARMS class for a Company ID.
5. Copy an existing ARMS class to a new or existing Company ID.

When the user selects to add or change an ARMS class another screen is displayed. This screen will allow the user to change or input an ARMS class, Enterprise class, class description and X12 car class.

When the user chooses to copy one or more ARMS classes, a screen will be displayed with the chosen ARMS classes, Enterprise classes, class descriptions and X12 car class, from which the user can choose a company ID, new or preexisting, to copy all of the classes to.

@Files:  
AMCLSTBL (C )  
A4CLSTBL (C )  
DCSPCL ( R )  
DCSPTY ( R )



<b>ARMS Process Report</b>
----------------------------

DCSPSH	( R )
DCSPAC	( R )
ARMSPR1	( R )
ARMSPR3	( R )

**Process**

Hierarchical numeric ID: 1.1.4.4.11

Coded name: AM0900

Name: SCR Create Invoice Request (AM0900)

Comment: @Definition:

This activity is performed by the help desk to generate paper and electronic bills for previously errored invoices.

## @Operational Method:

The user input fields include group, branch, and ticket number. AM0900 first verifies that the ticket is in the file, QRACCLSQ (closed ticket file). Secondly, it verifies that it is an ARMS ticket by checking a field in RACSLD. Finally AM0900 calls AM1010V1 that retrieves the corresponding AMXREF record.

Based on the status code returned, the program allows the user to proceed in AM0900. The next screen allows the user to hit a function key for re-billing and selecting of either paper or electronic billing method. A record is then written to RACBILL to reflect the rebill choice and an AMBILL record id written to log the rebill.

## @Notes:

- Runs on A,E,F,I,K,L,M,N,P,S,T,W,Y

## @Files: (CRUD)

-QRACCLSQ	( C )	(Logical over RACCLSC and past 12 monthly files)
-RACSLD	( R )	(Logical over RACSLC and past 12 monthly files)
-RACBILL	( C )	
-AMBILL	( C )	

**Process**

Hierarchical numeric ID: 1.1.4.4.12

Coded name: ARMS EDT

Name: SCR Display Reservation / Ticket Information (ARMS EDT)

Comment: @Definition: A process that provides the ARMS on-call staff the option of retrieving specific data elements from multiple Rental files.

## @Notes:

**Process**

Hierarchical numeric ID: 1.1.4.4.13

Coded name:

Name: ONL Monitor Purge Activity

Comment: @Definition:

This activity is used by ARMS on-call to monitor the progress of the ARMS application database purge.

**ARMS Process Report****Process**

Hierarchical numeric ID: 1.1.4.4.13.1  
Coded name:  
Name: ONL Display Database Transaction Set Controls  
Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

## @Notes:

- DSPDDTAARA AM006A
- Three steps involved in the entire purge process.

**Process**

Hierarchical numeric ID: 1.1.4.4.13.2  
Coded name:  
Name: ONL Display Orphaned Transaction Set Controls  
Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

## @Notes:

- DSPDDTAARA AM009A
- Three steps involved in the entire purge process.

**Process**

Hierarchical numeric ID: 1.1.4.4.13.3  
Coded name:  
Name: ONL Display Reorganize File Control  
Comment: @Definition: A process where by the ARMS on-call staff can monitor the database purge.

## @Notes:

- DSPDDTAARA AM009A
- Three steps involved in the entire purge process.

**Process**

Hierarchical numeric ID: 1.1.4.4.13.4  
Coded name:  
Name: MNU Display Purge Programs  
Comment: @Definition: see ONL Monitor Purge Activity.

**Process**

Hierarchical numeric ID: 0  
Coded name:  
Name: Rental and ARMS  
Comment: @Definition: This is the Automated Rental Management Systems (ARMS) application context as it currently is.

## On-Line Reporting - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjuster field.
- 3) Click **On-Line Reporting**.
- 4) The Select Report View Options screen appears (page 2).

The screenshot displays the 'Enterprise rent-a-car' web application interface. At the top, the 'Enterprise rent-a-car' logo is on the left, and a 'Home' link is on the right. Below the logo, the 'Office' field is set to 'B0020' and the 'Adjuster' field is empty. The main content area is divided into two columns of buttons. The left column, under the heading 'Rental Activity', includes buttons for 'Enter Last Name' (with 'KIN' entered), 'Create Reservation', 'Enterprise Requests by Adjuster', 'Change Reservation/Rental', and 'Enterprise Requests Unassigned'. The right column, under the heading 'Rental Lookup', includes buttons for 'View Reservation/Rental', 'On Line Reporting' (highlighted with a red border), 'Process Approved Invoices', and 'Setup and Maintenance'. A 'Logoff' button is located in the bottom left corner. The interface has a dark, textured background.

## Open Detail - By Adjustor

This Open Detail - By Adjustor screen appears when you select **Adjustor** in the **Report Sorted By** field and **Auth Days** in the **Additional Sort** field on the **Select Report View Options** screen (page 2).

- 1) Click to highlight the name of the renter whose file you want to access, and click **OK** to select that file.
- 2) The first **View Reservation/Rental** screen appears (**View Reservation/Rental**, page 4).

The following columns and fields are display-only:

**Adjustor Name** - The adjustor in charge of the claims listed below.

**Claim Number** - The claim number assigned to a particular renter.

**Type** - The type of claim (Insured, Claimant, Theft).

**Rate Authorized** - The rate of the vehicle authorized by the adjustor.

**Rent Days** - The number of days the vehicle has been on rent.

**Authorized Days** - The number of days authorized by the adjustor.

**Days Behind** - The difference (if any) between the number of days authorized and the number of days on rent.



If **Auth Days** was selected in the **Additional Sort** field, the claims will appear in descending order by the total authorized days.

**Enterprise**  
rent-a-car  
Office: B0020 01

**Open Detail**

Sort Order: Adjustor Last Name [ ] Search

Adjustor Name: ADAMS, JERRY

Claim Number	Type	Rate Authorized	Rent Days	Authorized Days	Days Behind	Rate	Days	Rate
1458334548	Insured	35.99	14	17		1		441.83
1459287311	Theft	36.99	15	16		1		591.84
1459246832	Claimant	20.99	15	13	Tern		Y	272.87
6578324681	Claimant	22.99	11	10		1		229.90
4519298216	Insured	20.99	12	8	Tern			167.92
8974656891	Insured	30.99	8	8	Tern			247.92
3589249521	Insured	20.99	12	8	Tern			167.92
1452121425	Claimant	26.99	4	7				188.93
4516824672	Insured	35.99	8	5		3	Y	179.95

Total Contracts: 9 Avg: 26.99 11.0 10.2 12.0 1.0 276.56

Back Excludes taxes/surcharges as applicable OK


## Open Detail - By Body Shop

This Open Detail - By Body Shop screen appears when you select **Body Shop** in the **Report Sorted By** field and **Auth Days** in the **Additional Sort** field on the **Select Report View Options** screen (page 2).

- 1) Key the first few letters of the desired body shop in the **Position to Body Shop Name** field, and click **Search**. The list refreshes with your selection at the top.
- 2) Click to highlight the name of the renter whose file you want to access, and click **OK** to select that file.
- 3) The first View Reservation/Rental screen appears (View Reservation/Rental, page 4).



For details on the column headings and fields, please refer to pages 3-4.



# Enterprise

rent-a-car

ARMS Automated Rental Management System

Open Detail

Office: B0020 01

By Body Shop:

Position to Body Shop Name:

Body Shop Name:

Contract #	Type	Rate	Days	Auth Days	Term	Auth	Days	Amount
1459287311	Theft	36.99	15	15		1		591.84
1459246832	Claimant	20.99	15	13	Term		Y	272.87
3216549875	Claimant	24.99	10	10				249.90
6578324681	Claimant	22.99	11	10	1			229.90
3589249521	Insured	20.99	12	8	Term			167.92
4516824672	Insured	35.99	8	5	3		Y	179.95

Total Contracts: 6

Avg: 27.15

11.8

10.3

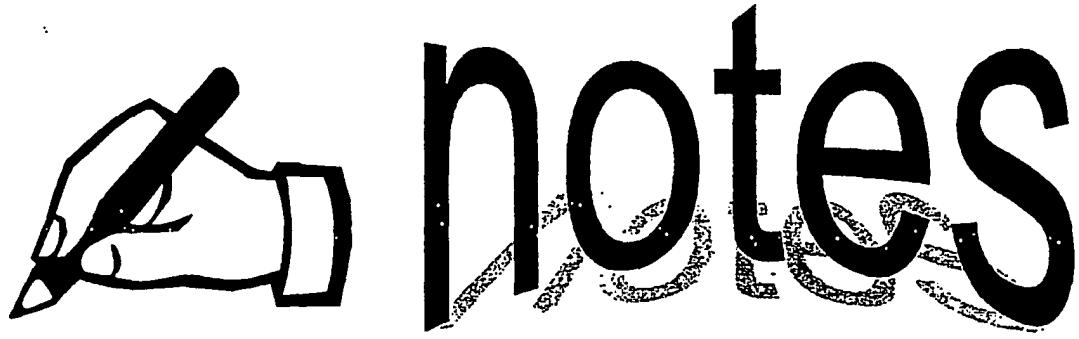
2.03

120

282.06

\*\*Excludes taxes/surcharges as applicable

Home Dynamic RC Java RC Show Detail



# Process Approved Invoices

## Process Approved Invoices



When should I use Process Approved Invoices?



Use this section to process and pay **all** invoices at your claims center.

Process Approved Invoices - Main Menu .....	1
Password Screen .....	2
Process Approved Invoices .....	3
Invoice Review/Approval .....	4
View Reservation/Rental .....	5
Send to Adjuster .....	6
Batch Payment Confirmation .....	7

## Process Approved Invoices - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjuster field.
- 3) Click Process Approved Invoices.
- 4) A password screen appears (page 2).

The screenshot shows the main menu of the Enterprise rent-a-car system. At the top, the logo "Enterprise rent-a-car" is displayed. To the right of the logo, there is a "Home" link. Below the logo, the "Office" field is set to "C0120" and the "Adjuster" field is set to "SUMNER". The main menu is divided into two columns of buttons. The left column contains buttons for "Rental Admin", "Create Reservation", "Enterprise Request by Adjuster", "Change Reservation/Rental", and "Enterprise Request Unassigned". The right column contains buttons for "Rental Lookup", "View Reservation/Rental", "On-Line Reporting", "Process Approved Invoices", and "Setup and Maintenance". A "Logoff" button is located at the bottom left of the screen.

Enterprise rent-a-car Home

Office: C0120 01 Adjuster: SUMNER

Handling by Adjuster: [ ]

Rental Admin

Enterprise Request by Adjuster

Change Reservation/Rental

Enterprise Request Unassigned

Rental Lookup

View Reservation/Rental

On-Line Reporting

Process Approved Invoices

Setup and Maintenance

Logoff



## Process Approved Invoices

This Process Approved Invoices screen appears when you enter the correct password on the Process Approved Invoices password screen (page 2).

- 1) Click to highlight the invoice you wish to view, and click OK to select the invoice.
- 2) The Invoice Review/Approval screen appears (page 4).
- 3) Click **Back** to return to the main menu (page 1).



If you click **Accept Batch**, the Batch Payment Confirmation screen appears (page 7).

You can also click either of the following option buttons:

**Home** - Return to the main menu (page 1).

**Refresh** - Remove all updated information from the current page.

**Enterprise rent-a-car** Office: C0120 01 Adjuster: SUMNER, CALI

Process Approved Invoices

Name	ID	Date	Location	Amount
JOAN	5825874510	10/22/99	SUMNER, CALI	146.03
MOORE, CAROLYN	5896247851	10/22/99	SUMNER, CALI	450.32
WATSON, PATTY	5698562541	10/22/99	SUMNER, CALI	300.00
WEBER, ANDREW	5698754821	10/22/99	SUMNER, CALI	511.76

Back Accept Batch OK

Home Refresh

## View Reservation/Rental

This View Reservation/Rental screen appears when you click View Rental on the Invoice Review/Approval screen (page 4).



*This screen is view-only. You can review the information on the screen to determine whether to process the adjustor-approved invoice.*

Click **Next** or **Back** to return to the Invoice Review/Approval screen (page 4).

You can also click either of the following option buttons:

**Home** - Return to the main menu (page 1).

**Additional Charges** - View additional charges on the contract (Appendix, page 7).

### Enterprise

rent-a-car

View Reservation/Rental

Office: C0120 01      Adjuster: SUMNER, CALI

Status: INVOICE RECEIVED      Type: INSURED      Claim #: 337517/5152

Renter: CALI (VIA)      ID: 104N

DATE	TIME	BY	DESCRIPTION	AMOUNT	CREDIT	DEBIT
10/10/99		S	INVOICE APPROVED FOR PAYMENT BY SUMNER, CALI			
10/16/99		R	INVOICE RECEIVED. AMOUNT DUE \$146.22		0	6
10/16/99		R	TICKET 073089 CLOSED 10/10/99 @ 14:56:00		0	6
10/16/99		S	AUTH CHANGE SENT BY SUMNER, CALI @ 14:55:52		0	6
10/16/99		S	EXTENSION SENT BY SUMNER, CALI @ 14:55:52		1	6

**Authorization Information**

Authorized Amounts:

Number of Days: 6

Vehicle Rate (Claimant):

Direct Bill Percent: 100 %

**Rental Location**

Policy Coverage (Insured Only):

Daily: 25.00

Maximum: 750.00

**Repair Information**

Rental Start: 10/05/99

Last Authorized Day: 10/00/00

Authorization Total: 150.00

**Renter Information**

Back
Next

Home
Additional Charges

## Batch Payment Confirmation

This Batch Payment Confirmation screen appears when you click **Accept Batch** on the Process Approved Invoices screen (page 3 or 6).

- 1) In the **Payment Date** field, enter the date you wish to make payment.
- 2) Click **OK** to process the payment and return to the main menu (page 1).  
Click **Back** to return to the Process Approved Invoices screen (page 3 or 6) without processing the payment.

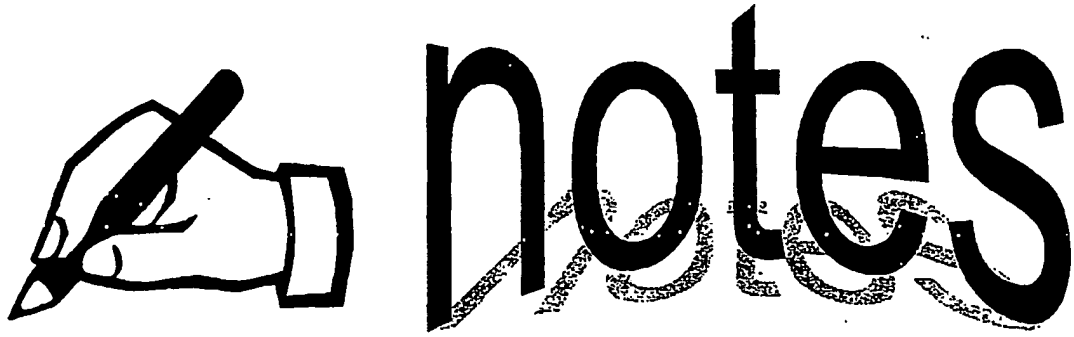


Be sure your office includes the reference number with payment.

The screenshot shows the 'Batch Payment Confirmation Screen' from the Enterprise Rent-A-Car system. The header includes the Enterprise logo, 'rent-a-car', and office information: 'Office: C0120 01' and 'Adjuster: SUMMER, CALI'. The screen displays the following fields and values:

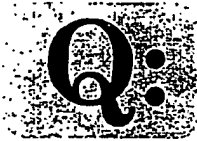
- Reference Number: 0001-00000000
- Payment Amount: 1-262.08
- Payment date: 02/10/2000
- Invoice Count: 1
- Remit payment to: ENTERPRISE RENT-A-CAR
- P.O. BOX 795110
- ST. LOUIS, MO 63179-0795

A note at the bottom states: 'Please include Reference Number with payment.' The screen has a 'Back' button at the bottom left and an 'OK' button at the bottom right. A vertical scrollbar is visible on the right side of the screen.

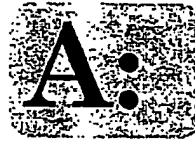


# Setup and Maintenance

## Setup and Maintenance



When should I use Setup and Maintenance?



Use this section to customize RMS Web, create or change passwords, and add and activate adjusters.

<b>Setup and Maintenance - Main Menu .....</b>	<b>1</b>
<b>Key Password to Access Setup and Maintenance - Main Menu .....</b>	<b>2</b>
<b>Choose a Selection from the File Maintenance Menu .....</b>	<b>3</b>
<b>Customer Profile Maintenance (Customize ARMS Web) .....</b>	<b>4</b>
<b>Adjuster Selection - Maintain Adjuster Files .....</b>	<b>5</b>
<b>Maintain Adjuster Files .....</b>	<b>6</b>
<b>Add an Adjuster .....</b>	<b>7</b>

## Setup and Maintenance - Main Menu

- 1) Key your claims office code in the Office field.
- 2) Key your last name in the Adjuster field.
- 3) Click Setup and Maintenance.
- 4) A password screen appears (page 2).



The *first* time you use the Setup and Maintenance screens, you will be prompted to create a password on the Customer Profile Maintenance screen (page 4). You can also change a password on the Customer Profile Maintenance screen. Once you have created a password, you will not be prompted on the Customer Profile Maintenance screen again.

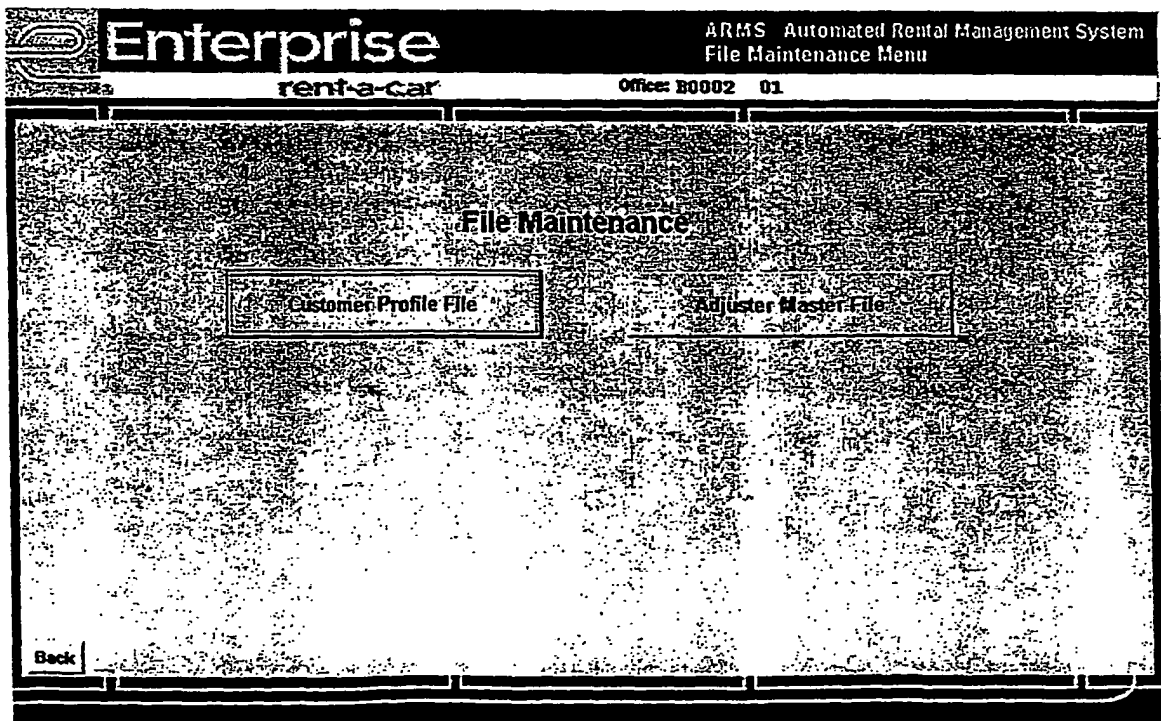
The screenshot shows the 'Enterprise rent-a-car' web application interface. At the top, there is a navigation bar with the 'Enterprise rent-a-car' logo and a 'Home' link. Below the logo, the text 'rent-a-car' is displayed. To the right of the logo, there are fields for 'Office: C0026' and 'Adjuster:'. Below these fields, there is a 'Handling for Adjuster' field with a dropdown menu set to '(Gr All)'. The main content area is divided into two columns. The left column contains a 'Rental Inquiry' section with a 'Renter's Last Name' field and a 'Create Reservation' button. Below this is an 'Enterprise Request by Adjuster' section with a 'Change Reservation/Rental' button and an 'Enterprise Requests Unassigned' button. The right column contains a 'Rental Lookup' section with a 'View Reservation/Rental' button and an 'On Line Reporting' button. Below this is an 'Invoicing' section with a 'Process Approved Invoices' button. At the bottom of the right column is an 'ARMS Maintenance' section with a 'Setup and Maintenance' button. A 'Logout' link is located in the bottom left corner of the main content area.

## Choose a Selection from the File Maintenance Menu

This File Maintenance Menu screen appears after the ARMS Web system accepts your password. There are two buttons from which to choose.

- ♦ **Customer Profile File** - Advance to the Customer Profile Maintenance screen to customize ARMS Web for your company's specific needs. Create or change your password (page 4).
- ♦ **Adjuster Master File** - Maintain adjuster information such as address and telephone number. You may also activate and deactivate an adjuster from the system (page 6).

- 1) Click the button (see descriptions above) you wish to choose.
- 2) Click **Back** to return to the main menu (page 1).



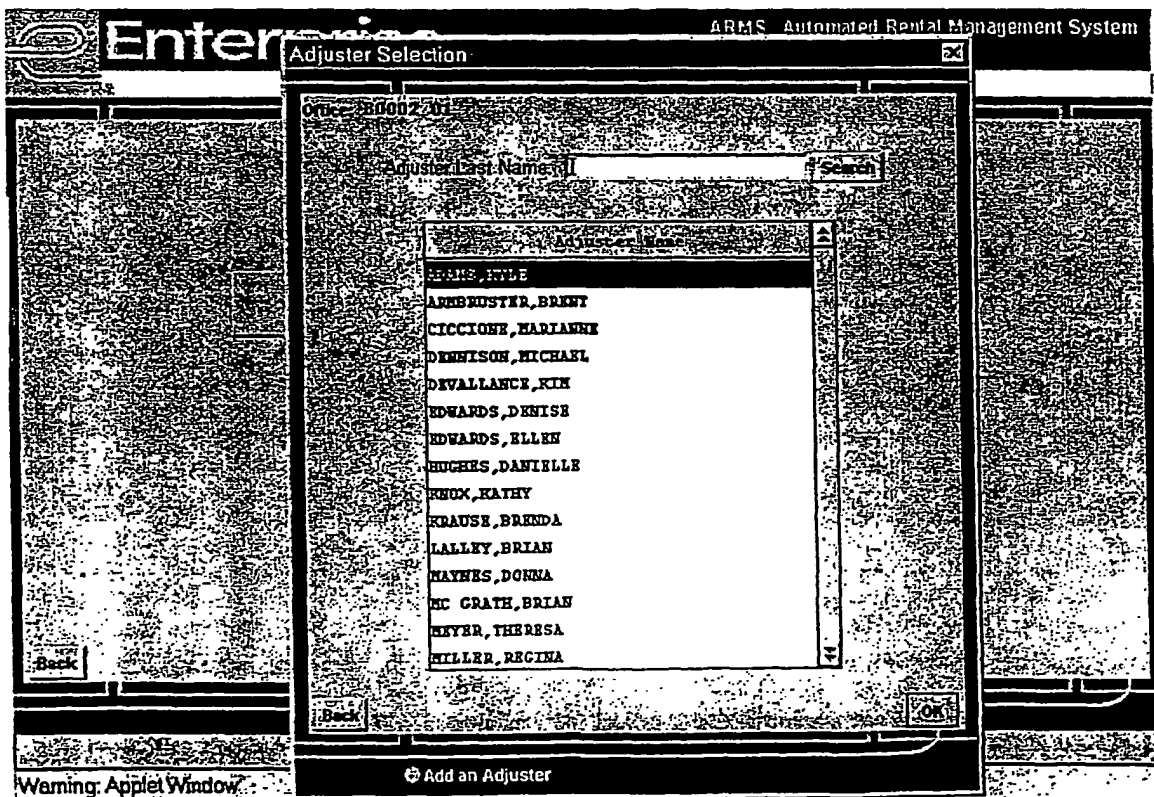
## Adjuster Selection - Maintain Adjuster Files

This Adjuster Selection pop-up window appears when you click **Adjuster Master File** on the File Maintenance Menu screen (page 3). There are two functions from which to choose.

- ◆ Maintain files for adjusters already authorized to ARMS Web.
- ◆ Add new adjusters to ARMS Web.

You can do one of the following:

- 1) Key the first few letters of a name in the **Adjuster Last Name** field, and click **Search** to find a particular adjuster.  
OR
- 2) Click to highlight the adjuster you wish to select, and click **OK** to continue to the Adjuster Maintenance screen (page 6).  
OR
- 3) Click **Add an Adjuster** to continue to the Adjuster Add screen (page 7).  
OR
- 4) Click **Back** to return to the File Maintenance Menu screen (page 3) without selecting or adding an adjuster.





## Add an Adjustor

This Adjuster Add screen appears when you click **Add an Adjustor** on the Adjustor Selection pop-up window (page 5).

- 1) Key the new adjustor's information on the appropriate lines.



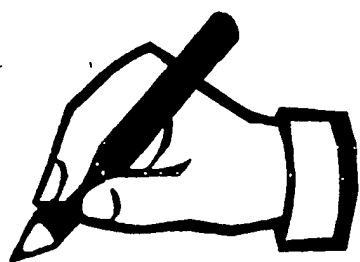
All information is *required* except the last four digits of a nine-digit ZIP code and a telephone extension.

- 2) Click **OK** to add the adjustor to the ARMS Web system and return to the File Maintenance Menu (page 3).  
Click **Back** to return to the Adjuster Maintenance screen (page 6).

You can also click the following option button (Appendix, page 1):

Home - Return to the main menu (page 1).

The screenshot shows the 'Enterprise rent-a-car' logo at the top left and 'ARMS Automated Rental Management System Adjuster Add' at the top right. Below the logo, it says 'Office: B0002 01'. The main form area contains several input fields: 'Adjuster Code', 'Last Name', 'First Name', 'City/State/Zip', 'Phone Number' (with 'Ext.' next to it), and 'Status' (with a dropdown menu showing 'Active'). At the bottom left, there are fields for 'Add Date/Time' and 'Chg Date/Time'. At the bottom right, there are 'Back' and 'OK' buttons. A 'Home' button is located at the very bottom left of the screen.



# notes



# Buttons

## Appendix



When should I  
use Buttons?



Use this  
section to  
access detail  
screens  
throughout  
ARMS Web.

Buttons - Overview .....	1
Rates .....	2
Detail .....	3
Message .....	4
Change Adjuster .....	5
Change Claims Office .....	6
Additional Charges .....	7

## Buttons - Overview

Buttons can guide you through the ARMS Web system quickly. This overview describes those buttons that are the same throughout each option. Screens that can be accessed *only* through a button are described on the following pages. Other buttons are defined within the options in which they are used.

**Rates** - View the rates, address, and phone number of the Enterprise office receiving the reservation.

**Home** - Return to the ARMS Web main menu.

**Detail** - Access screens to key or view more detail in a reservation/rental.

**Message** - Send a message to an Enterprise office.

**Change Adjuster** - Change the adjuster handling the file.

**Change Claims Office** - Change a claims office.

**Additional Charges** - View additional charges on a rental contract.

## Detail

### Additional Information in a Reservation Renter, Vehicle, Repair Facility, and Claim Information

This Create Reservation screen appears when you click Detail on the previous Create Reservation screen (Create a Reservation, page 2). It is also available in the Change Reservation/Rental and View Reservation/Rental sections.

- 1) Key the renter's Last Name, First name, and work and/or home telephone numbers (required).
- 2) Key any other information you wish to include about the renter, vehicle, repair facility, or coverage in the appropriate fields.
- 3) Click Next to save this information and continue to the next Create Reservation screen (below).



Any previously keyed information is retained and displays in the appropriate fields. This applies to both detail screens.



### Additional Information in a Reservation Authorization, Rental Delivery Information, and Message

This Create Reservation screen appears when you click Next on the previous Create, Change, or View Reservation/Rental screen.

- 1) If the Type is Insured or Theft, the Daily and Maximum fields under Policy Coverage and the Number of Days field must be filled in. Click Rates to view rates (optional). If the type is Claimant, the Number of Days and Vehicle Rate fields must be filled in. Key the Phone Number Where Rental Is Needed, and use the drop-down arrow to select Yes if the rental is needed immediately or No if not (required).
- 2) Key any other information you wish to include about the authorization, rental delivery, and message.
- 3) Click Finish to process the reservation and return to the ARMS Web main menu (Introduction, page v).

## Change Adjustor

Use this Adjustor Selection screen to look up or change an adjustor assigned to a reservation/rental.

- 1) Click to highlight the name of the adjustor you wish to select.  
To search for an adjustor by last name, key in the desired name and click **Search**.
- 2) Click **OK** to insert the adjustor's name in the appropriate fields. The name is locked in and carries forward to other screens of the authorization. Click **Back** to return to the previous screen without selecting an adjustor.



Click **Change Claims Office** to transfer the reservation/rental to another claims office. A pop-up window with a list of claims offices within your company displays (page 6).

**Enterprise** Adjuster Selection

Claim Number: [ ]

Authorization Information

Authorized Amount: [ ]

Number of Days: [ ]

Vehicle Rate (Claimant): [ ]

Direct Bill Partner: [ ]

NOT SPECIFIED

ECONOMY

COMPACT

Message: [ ]

Back

Rates Home

Warning: Applet Window

Adjuster Last Name: [ ] Search

DENNIS, KYLE

DENNISON, MICHAEL

DEVALANCE, KIM

KRAUSE, BRENDA

LALLEY, BRIAN

HAYNES, DONNA

MC GRATH, BRIAN

MEYER, THERESA

MILLER, REGINA

MONSON, DAVID

RANDOLPH, JANICE

SUMNER, CALI

TOTH, KELLY

UNKNOWN, UNKNOWN

AMS, KYLE

Claimant

Ext: 0000

Finish

Change Claims Office

## Additional Charges

Use this Additional Charges screen to authorize charges in addition to the price of the rental.

- 1) Click to select the type of surcharge in the Type column.  
Select from these three types:
  - ♦ Daily (the amount authorized is charged on a daily basis).
  - ♦ Rental (the amount authorized is charged once during the rental).
  - ♦ Percentage (the amount authorized is a percentage charge, e.g., airport access fee).
- 2) Key the dollar amount or percentage you are authorizing in the Amount or Percent column to the right of the Type column.
- 3) Click OK to save these charges and return to the previous screen. Click Back to return to the previous screen without authorizing any charges.



Once recorded, the information will be saved in the system but displays on this screen *only* when you click OK. Key over any existing information to make changes.

Enterprise

Create Reservation  
 Create Reservation

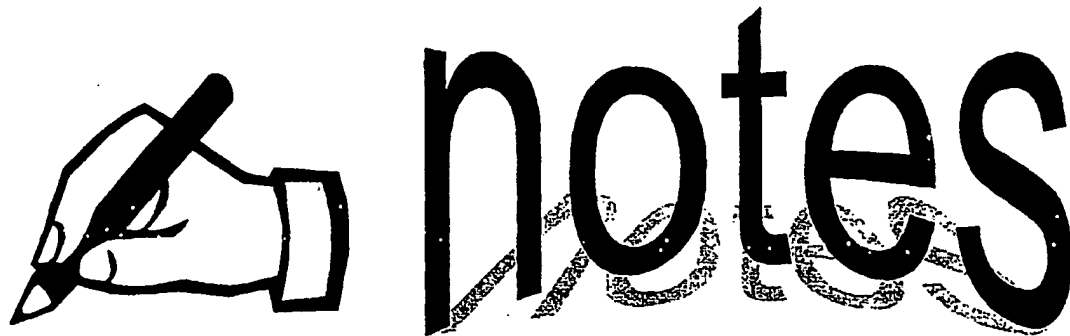
Office: B0003-01
Adjuster: ADAMS, KYLE

Charge	Type	Amount	Percent
CDW	Daily	8.99	
PAI			
MILEAGE CHARGE			
DPOI CHARGE			
MISC CHARGES			
UNDEFACE DRIVER			
SAFETY CAP SEAT			

\* Enter either an amount or percent, not both

Back
OK
Next

Message
Change Adjuster
Additional Charges





## ARMS/400 UPDATE

**Please be advised of the following update to the ARMS/400 program:**

ARMS/400 adjusters now have a new way to view open rentals. We have added a new feature that allows adjusters to sort claims by repair facility when searching for rentals by the last authorized day. This enhancement will enable adjusters to proactively manage their files (or the entire office's files) by consolidating all of the rentals that require follow-up by repair facility.

This is what the Select Reservation/Rental screen currently looks like after you select Option 2 - Change Reservation/Rental on the ARMS/400 Main Menu:

```

ARMS/400 - Automated Rental Management System
Select Reservation/Rental

Enter a name or claim number to position a list of
reservations/rentals for selection.

Adjuster Last Name: _____ First: _____
Claim Number: _____

To narrow your search, enter any of the additional information.

Adjuster Last Name: _____
Last Authorized Day: _____
Date of Loss: _____

Status:
1=Open Reservations
2=Open Rentals

Press Q to Quit or C to Continue
  
```

On the current screen, there is no way of sorting rentals by repair facility. The only sort available is by adjuster last name or the status of the rental.

## \*\* ARMS/400 UPDATE \*\*

### Updated Screen

The updated screen includes the field Sort by Repair Facility (see highlighted below).

```

ARMS/400 - Automated Rental Management System
Enter a name or claim number to position a list of
reservations/rentals for selection.

Enter Last Name: _____ First: _____
Claim Number: _____

To narrow your search, enter any of the additional information.
Approved Last Name: _____
Last Authorized Day: 01/01/01
Date of Loss: 01/01/01
Status: 1=Open Reservations
          2=Open Rentals

Sort by Repair Facility Y Y/N

F3=Exit F4=Lock F5=Previous
Press <ENTER> to Continue
  
```

All other filter options on this screen will remain the same. Please refer to your ARMS/400 User Guide for instructions on the other fields on this screen.

For this new feature, key the desired date in the Last Authorized Day field and Y (Yes) in the Sort by Repair Facility field. Press [ENTER]. Another new screen, Select Open Rentals by Repair Facility, appears (page 3).

## \*\* ARMS/400 UPDATE \*\*

### Select Open Rentals by Repair Facility Screen

```

ARMS/400 - Automated Rental Management System
Office: 0001 00
Select Open Rentals by Repair Facility

Position to Repair Facility:

Repair Facility: ABC Body Shop
11-01-11 14

Sel Repair Date Vehicle Days Auto Last
-----
1 WILLIAMS, DAN 1 87 HONDA ACCORD 14 14 11-01-11
2 STEPHENS, BOB 1 88 CHRY SUBURBAN 14 14 11-01-11
3 REYNOLDS, BOB 1 91 FORD EXPLORER 14 14 11-01-11
4 DAVIS, PAUL 1 89 CHRY CIRRUS 14 14 11-01-11
5 MILLER, TOM 1 88 ST PABLO 14 14 11-01-11
6 LEE, BOB 1 88 HONDA CIVIC 14 14 11-01-11
More...

F1=Exit F5=Previous Repairer F6=Next Repairer F12=Previous
  
```

The repair facilities appear in alphabetical order. To view a different repair facility, key the name in the Position to Repair Facility field and press [ENTER].

To select a file, key 1 in the Sel (Select) field and press [ENTER]. The Change Reservation/Rental appears (page 4).

- [F3] Exit  
Return to the ARMS/400 Main Menu.
- [F5] Previous Repairer  
Move back to the previous repair facility (in alphabetical order).
- [F6] Next Repairer  
Move to the next repair facility (in alphabetical order).
- [F12] Previous  
Return to the Select Reservation/Rental screen (page 2).

# ARMS/400 UPDATE

## Change Reservation/Rental Screen

The Change Reservation/Rental screen has also been updated (see highlighted below).

[illegible]

The phone number of the repair facility and the vehicle being repaired have been added. For detailed instructions on the other fields, please refer to your ARMS/400 User Guide.

*Note: These new fields will also be available on the View Reservation/Rental screen.*

**ARMS/400 UPDATE\*\***

Please be advised of the following update to the ARMS/400 program:

ARMS/400 is being enhanced with additions to Option 15 - On-Line Reporting. With these new features, more reports on rentals, broken down by adjuster, repair facility, or claim center, can be displayed on the screen.

Two new choices appear on the Select Report View Options screen:

- Closed in the Status field
- Summary in the Report Type field

The screen below is what the Select Report View Options screen currently looks like after Option 15 is selected on the ARMS/400 Main Menu.

```
ARMS/400 - Automated Rental Management System
*****
OBJECT: INCL 01      On-Line Reporting
*****
Select Report View Options

Report started by: 1 1=Adjuster, 2=Howling, 3=Office
Adjusted from: 1 1=Arbitrator, 2=Howling
Report Type: 1 1=Default
Status: 1 1=Open
*****
F12=The Screen
```

On the current screen, closed ticket reports or summary report types were not available.

## \*\*ARMS/400 UPDATE\*\*

### Updated Screen

```

ARMS/400 - Associated Railroad Management System
On-Line Reporting

Select Report View Options

Report Sorted By: 1 1=Adjuster 1=Report Facility 3=Closed

Status: 2 1=Open 2=Closed

Report Type: 2 1=Detail 2=Summary

Additional Sort: 1 1=Sort Days 2=Sort Length

F12=Previous
  
```

From the updated Select Report View Options screen, adjusters can now select summary reports and reports on closed tickets — in addition to all of the other reports they could previously access with Option 15.

➤ *The fields still automatically default to 1. You must key 2 to access the new choices on the menu.*

1. Key the desired sorting method.
  2. Press [ENTER].  
 If 2 is keyed in the Status field and 1 is keyed in the Report Type field, a Closed Detail screen appears (page 3).  
 If 2 is keyed in the Status field and 2 is keyed in the Report Type field, a Closed Summary screen appears (page 5).
- *Note: These are the only two screens that will be shown in this update. Both will be sorted by Adjuster and Authorized Days. For instructions on other screens, please refer to the ARMS/400 User Guide.*

## \*\* ARMS/400 UPDATE \*\*

### Closed Detail Screen

ARMS/400 - Automated Rental Management System										
Closed Detail - By Adjuster - By Auth Days										
Position to Adjuster Last Name:				Office Active		Range				
Adjuster Name:				Adjuster Name		Range				
Seq	Claim Number	Type	Rate	Rent	Auth	# of	Total	Amt	Total	
			Auth	Days	Days	Ext	Chgs	Rcvd	Amount	
1	0000000000	I	25.00	30	30	0	150.00	600.00	600.00	
2	0000000000	I	25.00	20	20	2	100.00	400.00	400.00	
3	0000000000	I	25.00	15	15	0	75.00	300.00	300.00	
4	0000000000	I	25.00	10	10	0	50.00	200.00	200.00	
5	0000000000	I	25.00	5	5	0	25.00	100.00	100.00	
Total:			Avg:		25.00	10.6	14.2	1.6	541.00	220.00
More										
F0=Exit F1=Prints Adj F2=Next Adj F3=Show Header F4=Menu										

From the Closed Detail screen, adjusters can select an individual claim or simply look at the information provided on this screen.

1. **Position to Adjuster Last Name** - Key the first few letters of the last name of the adjuster whose claims you wish to view.
2. Press **[ENTER]** to display the selected adjuster's name.

The following fields are display-only (unless noted below):

**Office Active** - The first month with available reporting data.

**Range** - The range of dates for the reporting data.

- *These are two new fields on the screen. They are both prefilled in with default values. You can edit the Range field, but the Office Active field is view-only.*

**Adjuster Name** - The adjuster in charge of the claims listed below.

**Claim Number** - The claim number assigned to a particular renter.

**Typ** - The type of claim (I=Insured, C=Claimant, T=Theft).

**Rate Auth** - The rate of the vehicle authorized by the adjuster.

**Rent Days** - The number of days the vehicle has been on rent.

**Auth Days** - The number of days authorized by the adjuster.

**# of Ext** - The number of extensions the adjuster has sent.

**Total Chgs** - The total amount of charges on the rental contract.

**Amt Rcvd** - The amount of the payment received by Enterprise.

**ARMS/400 UPDATE**

**Bill Amount** - The amount of the rental bill.

**Total** - The number of contracts the adjuster is handling at this time.

**Avg** - The average numbers from the columns above.

3. Key 1 in the Sel (Select) field to choose a particular claim.
4. Press [ENTER]. The View Reservation/Rental screen appears. Please refer to the ARMS/400 User Guide to view the View Reservation/Rental screen.

- |       |   |
|-------|---|
| [F3]  | <b>Exit</b><br>Return to the Main Menu (page 1).  |
| [F5]  | <b>Previous Adj</b><br>Display the previous adjuster's (alphabetically) claims (if applicable). |
| [F6]  | <b>Next Adj</b><br>Display the next adjuster's claims (if applicable).                          |
| [F9]  | <b>Show Renter</b><br>Display renters' names in place of the claim numbers.                     |
| [F12] | <b>Prev</b><br>Return to the Select Report View Options screen (page 2).                        |



## \*\*ARMS/400 UPDATE\*\*

### Closed Summary Screen

ARMS/400 - Automated Rental Management System  
Closed Summary - By Adjuster

Position (1) Adjuster Last Name: \_\_\_\_\_ Office Active: 01/19/99  
Range: 01/01/99-01/31/99

Adjuster Name	Tot Inv	Rate Auth	Rent Days	Auth Days	# of Ext	Total Chgs	Amt Rcvd	Bill Amount
WALSH, JOHN	10	24.00	25.0	24.0	1	54.00	450.00	546.00
DE VRIES, JIM	40	24.00	21.5	21.5	1	48.00	411.00	502.00
ELSON, JIM	10	24.00	11.7	11.3	1	42.00	418.00	480.00
STANLEY, JIM	20	24.00	17.3	17.0	1	42.00	373.00	459.00
WELSH, JOHN	22	24.00	10.1	9.9	1	32.00	273.00	315.00
<b>Summary</b>	<b>104</b>	<b>24.00</b>	<b>104.1</b>	<b>103.8</b>	<b>5.2</b>	<b>542.00</b>	<b>546.00</b>	<b>546.00</b>

Page 1 of 1

This Summary screen contains average totals for adjusters.

**Position to Adjuster Last Name** - Key the first few letters of the last name of the adjuster whose claims you wish to view.

The following fields are display-only (unless noted below):

**Office Active** - The first month with available reporting data.

**Range** - The range of dates for the reporting data.

- *These are two new fields on the screen. They are both prefilled in with default values. You can edit the Range field, but the Office Active field is view-only.*

**Adjuster Name** - The adjuster in charge of the claims.

**Tot Inv** - The total number of invoices for this adjuster.

**Rate Auth** - The rate of the vehicle authorized by the adjuster.

**Rent Days** - The number of days the vehicle has been on rent.

**Auth Days** - The number of days authorized by the adjuster.

**# of Ext** - The number of extensions the adjuster has sent.

**Total Chgs** - The total amount of charges on the rental contract.

**Amt Rcvd** - The amount of the payment received by Enterprise.

**Bill Amount** - The amount of the rental bill.

**Summary** - The average of all of the columns above.

[F3]

Exit

Return to the Main Menu (page 1).

# ARMS/400 UPDATE

[F12]

Previous

Return to the Select Report View Options screen (page 2).

\* \* \*

---

# **Enterprise Rent-A-Car**

## **ARMS Web 3.0 Functional Design Specification Extend Rental**

**Version 1.1**  
**Last Saved: 9/28/00 2:55 PM**

**EXHIBIT E**

ARMS Redesign Project – Release 3.0  
Extend Rental

Issue: 1.1  
Issue Date: 10/20/00

## Revision History

Date	Issue	Description	Author
April 1, 2000	0.1	Created Use Case and Screen Design Documents	Keith Baker, Debi Ealick, and Johnny Sands
April 10, 2000	0.1	Linked subdocuments to master document	Cindy Bastean
May 3, 2000	0.2	Removed subdocuments and formatted according to standards	Cindy Bastean
May 16, 2000	0.3	Incorporated changes from cross team QA	Cindy Bastean
May 31, 2000	0.4	Changed screen information	Cindy Bastean
June 8, 2000	0.4	Added data field information	Cindy Bastean
July 3, 2000	0.5	Updated screen information	Cindy Bastean
July 13, 2000	0.5	Updated use case and screen design sections	Mike Slater, Brian Weingart, Johnny Sands, Debi Ealick, Brent Armbruster and Cindy Bastean
September 25, 2000	1.1	Changes made based on feedback provided by business leads with respect to the future state of Release 3.0.	Amanda Banta, Aaron Foster, Mike Slater, Tim Weinstock

ARMS Redesign Project – Release 3.0

Issue: 1.1

Extend Rental

Issue Date: 10/20/00

## Table of Contents

1.	Extend Rental Use Case	1
1.1	Application Overview	1
1.2	Brief Description	1
1.3	Use Case Actors	1
1.4	Pre-Conditions	1
1.5	Flow of Events	1
1.5.1	Activity Diagram	2
1.5.2	Basic Flow	3
1.5.3	Alternative Flows	3
1.6	Post-Conditions	4
1.7	Special Requirements	4
1.8	Extension Points	4
1.8.1	<u>MA-16 Reassign USER/Office</u> (Transfer)	4
1.8.2	<u>MA-08 View Car Class</u>	4
1.8.3	<u>MA-15 Terminate Rental</u>	4
1.8.4	<u>MA-04 Send Message</u>	4
2.	Screen Design	5
2.1	Extend Rental Detail	5
2.1.1	Screen Layout - Extend Rental Detail	5
2.1.3	Extend Rental Detail	10
2.1.4	Screen Function Definition	11

## Extend Rental

### 1. Extend Rental Use Case

#### 1.1 Application Overview

The following is a document used to illustrate the process for how the USER will extend a previously authorized rental using ARMS/Web 3.0. The intent for this release of the ARMS/Web application is to reach a much wider audience. This application will target a Multi-Vendor, Multi-Segment, and International customer base.

#### 1.2 Brief Description

This use case will describe how the USER will extend a previously authorized rental. The rental company (via an Authorization Request), the RENTAL ADMINISTRATOR (via a Customer Search), or Reporting (via the Callback feature) can initiate this use case.

#### 1.3 Use Case Actors

The following actors will interact with this use case:

- **RENTAL ADMINISTRATOR** – The RENTAL ADMINISTRATOR will use the system to extend a previously authorized rental. This use case refers to a USER in the role of a rental administrator. There are various types of customers that the USER would represent, which include corporate account holders, car dealerships, insurance companies, and others.
- **ARMS** – The ARMS system will receive/send transactions to ARMS/Web to confirm the extended rental.
- **RENTAL CAR COMPANY** – A wide variety of rental car companies will be able to use this system as well. Each company will have the ability to initiate and manage their rentals through the use of this application.

#### 1.4 Pre-Conditions

- The USER must have logged into the ARMS/Web system.
- The USER must have selected a previously authorized, open rental.

#### 1.5 Flow of Events

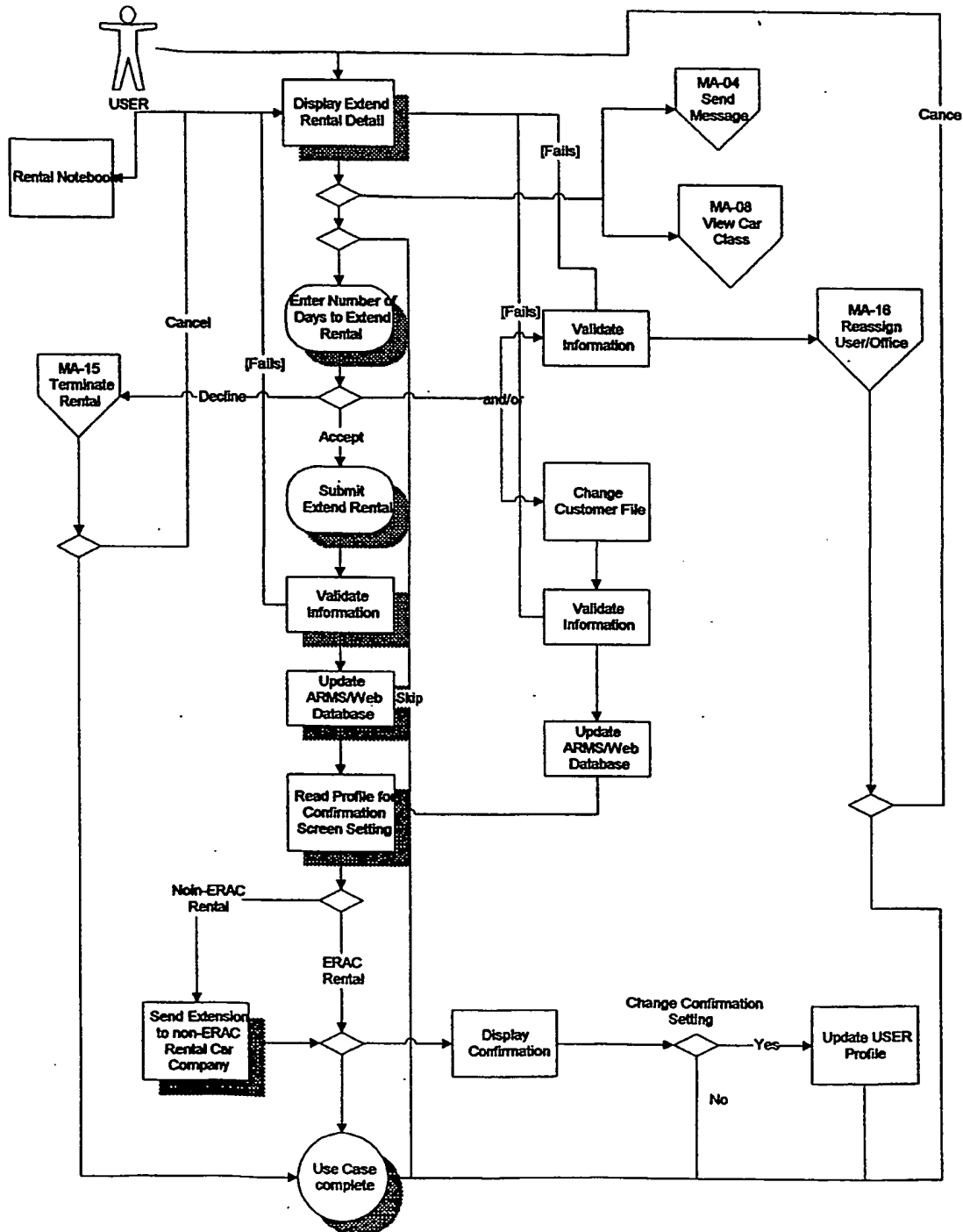
The Flow of Events will include the necessary steps to make changes and updates to "Extend Rental".

ARMS Redesign Project – Release 3.0  
Extend Rental

Issue: 1.1  
Issue Date: 10/20/00

### 1.5.1 Activity Diagram

Extend Rental Activity Diagram



ARMS Redesign Project – Release 3.0

Issue: 1.1

Extend Rental

Issue Date: 10/20/00

---

### 1.5.2 Basic Flow

1. The system will display the details of the Rental.
2. The USER will enter the number of days to extend the rental.
3. The USER will submit the Extend Rental Details.
4. The system will validate the number of days the rental will be extended.
5. The system will update the ARMS/Web database with the Extend Rental Details.
6. The system will read the profile for the confirmation screen setting.
7. For non-Enterprise rentals, the extension is sent to the non-ERAC rental car company's rental system.
8. This ends the use case.

### 1.5.3 Alternative Flows

#### 1.5.3.1 View Rental Notebook

At step 1 of the basic flow, the USER may choose to view the history of a rental. The USER will be able to see the diary notes associated with the Reservation / Rental.

#### 1.5.3.2 Display Confirmation

After step 7, the USER may wish to have a confirmation page displayed, indicating that some type of change has taken place. The confirmation page is completely optional; therefore, at anytime the USER wants to set their profile to bypass this screen, he/she may do so.

#### 1.5.3.3 Update USER Profile

During the confirmation process, the USER has the option of changing their profile setting to display or hide the confirmation page. Each time the setting is changed, the USER profile must be updated to reflect the new requirements set by the USER.

#### 1.5.3.4 Validate Changes

If the USER changes or adds information, which does not pass validation, an error message will notify the USER and return them to step 1 of the Basic Flow.

If an error is discovered in the validation of the reservation / rental information submitted by the USER, the system would present the USER with an error message and return them to the Detailed Reservation / Rental Display. If the error is specific to a data field within the form, the field should be highlighted and the error described.

#### 1.5.3.5 Change Customer File

Prior to step 3, the USER has the option to make changes to the customer file. After clicking the change/add link, the screen will refresh with all editable fields opened and available for the USER to make changes.

#### 1.5.3.6 Update ARMS/Web Database

After successfully validating the recent changes, the system must update the ARMS/Web Database. The system goes through the same process as in the Basic Flow, as the database is updated to reflect the latest changes.



**1.6 Post-Conditions**

- If the use case was successful then the rental has been extended and the ARMS/Web system has been notified.
- If the use case was unsuccessful then the system has remained unchanged.

**1.7 Special Requirements**

- The number of days to extend a rental must be an integer greater than zero.
- If a USER attempts to extend an insured rental beyond their limits for number of days and dollar amount, the system should return an error message.

**1.8 Extension Points****1.8.1 MA-16 Reassign USER/Office (Transfer)**

After the extend rental detail is displayed, the USER may choose to transfer the current office/USER. First, the USER would select to change the current office/USER. Second, the system would display a list of authorized offices/USERS. Third, the USER would select a new office/USER. If additional changes are made to the customer file, the new data will also be passed through the transfer process.

**1.8.2 MA-08 View Car Class**

The View Car Class use case will be used to allow the USER to view details about and select a car class to apply to a reservation. Details will include the average number of passengers and luggage items that can be served by a vehicle in the specific car class. The car class selected by the USER should be applied to the reservation.

**1.8.3 MA-15 Terminate Rental**

After the extend rental detail is displayed, the USER may choose to terminate the rental. If termination is selected, the USER must enter a reason for the termination of the rental. Termination means the insurance company is no longer willing to pay for the rental.

**1.8.4 MA-04 Send Message**

The Send Message will be used to allow the USER to capture messages and diary notes associated with extending a rental. The USER can elect to either have the message sent to the rental company responsible for the reservation/authorization, or (Depending on the user segment if this option is available) to store the note in the ARMS/Web system without sending the message to rental company. All MESSAGES and DIARY NOTES captured must be related to a specific reservation/authorization.

ARMS Redesign Project – Release 3.0  
Extend Rental

Issue: 1.1  
Issue Date: 10/20/00

## 2. Screen Design

A definition of the screen layout(s), screen data fields, and screen functions that are used to implement the flows identified above. More than one screen may be used to implement support for the use case flow.

### 2.1 Extend Rental Detail

This screen will allow the USER to pick which functions that he/she may want to change.

#### 2.1.1 Screen Layout - Extend Rental Detail

(ARMS/Web 2.0)

**Extend Rental**  
CUSTOMER FILE

2 of 3 Action Items

**RENTAL INFORMATION**

Vehicle: Compact V10  
Start Date: 03/20/2000  
End Date: 03/27/2000  
Start Time: 08:00 AM  
End Time: 08:00 AM  
Rate: \$21.99/day  
Additional Charges: None  
Total: \$21.99

**ADDITIONAL CLAIM INFORMATION**

Claim Number: 225217502022123  
Claim Type: Theft  
Insured Driver: Latamarcia, Cruz  
Driver's License: CHIC 5444444 1599  
Date of Loss: 03/25/2000  
Loss Type: Non-Owner  
Police: Daily rate  
Recovery: \$0.00

**RENTAL INFORMATION**

Authorized Class: Standard  
Buyer/Renter: 3 days @ \$21.99/day  
Current Class: Full-Size  
Additional Charges: None  
Rental Rate: \$21.99/day  
Start Date: 03/20/2000

**Enterprise Rent-A-Car locations**

Enterprise Rent-A-Car locations  
Enterprise Rent-A-Car locations  
5400 N. Ashland  
Chicago, IL 60622  
773.334.6400

**REPAIR FACILITY:**  
Elco Chevrolet  
Chicago, IL 60621  
773.334.6400

**MESSAGE:**  
Message: Chicago, Illinois, 20000  
Rate: 2000 Enterprise, 20000, 20000  
Extension Request: 20000  
Extension: 20000

© 2000 ARMS

Confidential | Terms & Conditions

Confidential

Enterprise Rent-A-Car 2000

5

Y:\APPS\ARMSWeb\Application\ARMSWEBV3\@db\Functional Specs\MA-12 Extend Rental\Functional Specs\MA-12-FDS-ARMSWeb3.0v1.1.doc

ARMS Redesign Project - Release 3.0

Extend Rental

Issue: 1.1

Issue Date: 10/20/00

(Insurance User)

(Change or Add)

## RENTER INFORMATION:

Bowle, David  
1735 N. Paulina St.  
Chicago, IL 60622

Home: (773)564-6054  
Work: (773)395-6200  
Email: dbowle@zefer.com  
Requested email confirmation

## RENTAL INFORMATION:

Authorized Class: Standard  
Days/Rate: 5 days @ \$21.99/day  
Current Class: Full-Size  
Additional Charges: None  
Direct Bill %: None  
Rental Date: 03/28/2000  
Start Date: 03/20/2000

Rental Location:  
Enterprise Edgewater Branch  
5400 N. Ashland  
Chicago, IL 60622  
773-394-5400

## ADDITIONAL CLAIM INFORMATION:

Claim Number: 323232323232323  
Claim Type: Theft  
Insured Name: Latumandier, Craig  
Owner's vehicle: GMC Suburban 1999  
Date of Loss: 03/28/2000  
Loss Type: Non-Drivable  
Policy: Daily rate/  
Maximum dollars: 30,500

Repair Facility:  
Elco Chevrolet  
Chicago, IL 60621  
(773)334-9832

## NOTES/COM:

Message, Belanger, Hugues, 2/20/00  
Note from Enterprise, Sarussi, Marty, 2/21/00  
Extension Request, 2/24/00  
Extension, 2/25/00

◆ top of page

Contact Us | Terms & Conditions | Log Off

Confidential

Enterprise Rent-A-Car © 2000

Y:\APPS\ARMSWeb\Application\ARMSWEB\3\@db\Functional Specs\MA-12 Extend Rental\Functional Specs\MA-12-FDS-ARMSWeb3.0v1.1.doc

6

ARMS Redesign Project – Release 3.0

Extend Rental

Issue: 1.1

Issue Date: 10/20/00

(Fleet User)

Welcome to the  
**Automated Rental Management System**

Create a RESERVATION Find a CUSTOMER

Office: 001 Handling for: Yourself

You just authorized 3 days at \$29.39/day for Hooks, Tom

**Extend Rental:** for Bowie, David Claim no. 765849322-001  
CUSTOMER FILE

2 of 4 Action Items

Extend Rental Request for:

☐ Additional Authorized Days for: **Compact/21.95** **21/500**

Message:  
06/10/00 08:25 / more days. Working on Claim.  
06/08/00 Working on Claim.  
06/07/00 Extension processed through 06/08/00.  
7-day extension requested.  
Data Modified.

Rental Location:  
Enterprise Edgewater Branch  
773-334-5400

Current Rental Status:  
Rental Start Date: 01/01/00  
Last Authorized Date: 06/10/00  
Authorized to Date: 06/08/00  
Charges to Date: \$229.00

Repair Facility:  
Elco Chevrolet  
773-334-9832  
Owner Vehicle: 1999 GMC Suburban

☐ Return this rental?

**[Change or Add]****RENTER INFORMATION:**Bowie, David  
1735 N. Pauline St.  
Chicago, IL 60622Home: (773)554-6054  
Work: (773)395-6200  
Email: dbowie@zefer.com  
Requested email confirmation**RENTAL INFORMATION:**Authorized Class: Standard  
Days/Rate: 5 days @ \$21.99/day  
Current Class: Full-Size  
Additional Charges: None  
Rental Date: 03/28/2000  
Start Date: 03/20/2000**Rental Location:**  
Enterprise Edgewater Branch  
5400 N. Ashland  
Chicago, IL 60622  
773-334-5400**ADDITIONAL CLAIM INFORMATION:**Claim Number: 323232323232323  
Claim Type: Theft  
Insured Name: Lalumandier, Craig  
Owner's vehicle: GMC Suburban 1999  
Date of Loss: 03/28/2000  
Loss Type: Non-Driveable  
Policy: Daily rate/  
Maximum dollars: 30/600**Repair Facility:**  
Elco Chevrolet  
Chicago, IL 60621  
(773)334-9832**NOTEBOOK:**Message, Belanger, Hugues, 2/20/00  
Note from Enterprise, Sarussi, Marty, 2/21/00  
Extension Request, 2/24/00  
Extension, 2/25/00

• top of page

Contact Us | Terms &amp; Conditions | Log On

Confidential

Enterprise Rent-A-Car © 2000

7

Y:\APPS\ARMSWeb\Application\ARMSWEBVJ\@dib\Functional Specs\MA-12 Extend Rental\Functional Specs\MA-12-FDS-ARMSWeb3.0v1.1.doc

ARMS Redesign Project – Release 3.0

Extend Rental

Issue: 1.1

Issue Date: 10/20/00

(Dealership User)

Welcome to the  
**Automated Rental Management System**

**GROUP A - RESERVATION** **Find a CUSTOMER**

Office: 001      Handling for: Yourself

You just authorized 3 days at \$29.39/day for Hanks, Tom

**Extend Rental:** for Bowie, David Purchase Order No.765849322-001  
CUSTOMER FILE

2 of 4 Action Items

Customer requested for:

☐ **Extend Rental for 3 days @ Compact/21.95** **Policy Limit: 20/500**

Messages:

03/20/00 05:11 3 more days - Making an Error  
03/20/00 05:11 Making an Error  
03/20/00 Extension requested through 03/28/00  
3 days extension requested  
On 3/20/00

Current Rental Status:

Rental Start Date: 03/25/00  
Rental Authorization: 04/13/00  
Authorized by User: 5 days  
Charges to Date: \$229.00

Rental Location:

Enterprise Edgewater Branch  
773-334-5400

Repair Facility:

Elco Chevrolet  
773-334-9832  
Current Vehicle: 1999 GMC Suburban

☐ **Extend this rental?**

**CONFIRMATION NUMBER: 765849322**

[Change or Add]**RENTER INFORMATION:**Bowie, David  
1735 N. Paulina St.  
Chicago, IL 60622Home: (773)564-6054  
Work: (773)395-6200  
Email: dbowie@zefer.com  
Requested email confirmation**RENTAL INFORMATION:**Authorized Class: Standard  
Days/Rate: 5 days @ \$21.99/day  
Current Class: Full-Size  
Additional Charges: None  
Rental Date: 03/28/2000  
Start Date: 03/20/2000**Rental Location:**  
Enterprise Edgewater Branch  
5400 N. Ashland  
Chicago, IL 60622  
773-334-5400**ADDITIONAL CLAIM INFORMATION:**Purchase Order Number: 32323232323  
Bill Type: Theft  
Insured Name: Lalumandier, Craig  
Owner's vehicle: GMC Suburban 1999  
Date of Loss: 03/28/2000  
Loss Type: Non-Driveable  
Policy: Daily rate/  
Maximum dollars: 30/500**Repair Facility:**  
Elco Chevrolet  
Chicago, IL 60621  
(773)334-9832**NOTESBOX:**Message, Belanger, Hugues, 2/20/00  
Note from Enterprise, Sarussi, Marty, 2/21/00  
Extension Request, 2/24/00  
Extension, 2/25/00[top of page](#)[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

Confidential

Enterprise Rent-A-Car © 2000

8

Y:\APPS\ARMSWeb\Application\ARMSWEBV3\ddb\Functional Specs\MA-12 Extend Rental\Functional Specs\MA-12-FDS-ARMSWeb3.0v1.1.doc

ARMS Redesign Project – Release 3.0

Issue: 1.1

Extend Rental

Issue Date: 10/20/00

(Corporate User)

Welcome to the  
Automated Rental Management System

Create a RESERVATION Find a CUSTOMER

Office: 001 Handling for: Yourself

You just authorized 3 days at \$29.39/day for Hanks, Tom

**Extend Rental:** for Bowie, David Corporate Class No. 765849322-001  
CUSTOMER FILE

2 of 4 Action Items

Extended requested for:

☐ Enterprise Full-Size 2000 **Compact/21.95** ☐ 2000-2000

Policy Limit: **20/500**

Message:  
08/11/00 BSC 2 hours ago: Waiting on Data  
08/11/00 Waiting on Data  
08/21/00 Extension requested through 10/26/00  
1 day extension requested  
Guth, Nathan

Message to Call:

Rental Extension:  
Enterprise Edgewater Branch  
773-334-5400

Current Rental Status:

Rental Class Date:	5/15/00
Class Authorized Date:	06/13/00
Authorized to Date:	5 days
Changes to Date:	6/23/00

☐ Extend this rental?

Press ctrl+enter to save and work again

[\[Change or Add\]](#)**RENTER INFORMATION:**

Bowie, David  
1735 N. Paulina St.  
Chicago, IL 60622

Home: (773)564-6054  
Work: (773)395-6200  
Email: dbowie@refer.com  
Requested email confirmation

**RENTAL INFORMATION:**

Authorized Class: Standard  
Days/Rate: 5 days @ \$21.99/day  
Current Class: Full-Size  
Additional Charges: None  
Rental Date: 03/28/2000  
Start Date: 03/20/00

Rental Location:  
Enterprise Edgewater Branch  
5400 N. Ashland  
Chicago, IL 60622  
773-334-5400

**ADDITIONAL CLASS INFORMATION:**

Corporate Class Number: 3232323232323  
Lease Type: Non-Drivable  
Policy: Daily rate/  
Maximum dollars: 30/500

**NOTEBOOK:**

Message, Belanger, Hugues, 2/20/00  
Note from Enterprise, Sarussi, Marty, 2/21/00  
Extension Request, 2/24/00  
Extension, 2/25/00

[top of page](#)[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

Confidential

Enterprise Rent-A-Car © 2000

9

Y:\APPS\ARMSWeb\Application\ARMSWEB\3\@db\Functional Specs\MA-12 Extend Rental\Functional Specs\MA-12-FDS-ARMSWeb3.0v1.1.doc

ARMS Redesign Project – Release 3.0  
 Extend Rental

Issue: 1.1  
 Issue Date: 10/20/00

### 2.1.3 Extend Rental Detail

Screen Label	Type	Size	Screen Field Name	Data Field Name	Screen Specific Rule
Additional Charges	Output	15	Additional Charges		
Handling For:	Output	30	Handling for Adjuster's Name	First Name + Last Name	Last Name + First Name
Note to Self Only	Input	50	Message	NOTE	
Messages:	Output	8	Message Creation Date	Add Date	N/A.
Note to Enterprise:	Input	50	Message Text	NOTE	N/A.
	Output	50	Message Text	NOTE	N/A.
Claim Number: Purchase Order Number Corporate Class Number	Output	11	Claim Number Purchase Order Number Corporate Class Number	Insurance Claim Number, PO#, CC#	
Days Authorized to Date:	Output	2	Number of Days Authorized	Number Of Days Authorized	N/A.
__ additional authorized days	Output	2	Number of Days to Extend	Number of Days to Extend	
Policy Limits	List Box	5	Policy Maximum and Dollars per day	Max \$ Covered + Dollars Per Day Covered	
	Output	30	Rental Location Branch Name	Rental Location	
days @:	List Box	6	Rental Location Rate	Vehicle Rate	N/A.
Date of Rental	Output	10	Rental Start Date	Start Date	N/A.
Insured Name:	Output	30	Insured's Name	First Name + Last Name	
	Output	30	Rental Location Address	Address Line + Address Line2	N/A.
	Output	25	Rental Location City Name	City	N/A.
	Output	10	Rental Location Postal / Zip Code	Zip Code	N/A.
	Output	3	Rental Location State / Province Code	State	N/A.
	Output	13	Rental Location Telephone Number	Telephone Number	N/A.
Date of Loss:	Output	10	Date of Loss	Date Of Loss	
	Output	20	Renter City Name	City	
	Output	10	Renter Postal / Zip Code	Zip Code	
	Output	3	Renter State / Province Code	State	
	Output	30	Renter Street Address	Address Line	
Home:	Output	16	Renter's Home Phone	Renters Night Phone + Renters Night Phone Extensin	Not editable if ticket is Open.

ARMS Redesign Project – Release 3.0  
Extend Rental

Issue: 1.1  
Issue Date: 10/20/00

Screen Label	Type	Size	Screen Field Name	Data Field Name	Screen Specific Rule
	Output	30	Renter's Name	First Name + Last Name	Will not be editable if ticket is open. First Name + Last Name
Renter Information:	Output	30	Renter's Name	First Name + Last Name	N/A.
Work Phone:	Output	16	Renter's Work Phone	Day Phone + Renters - Day Phone Extension	Will not be able to edit if ticket is Open.
Owner's vehicle:	Output	4	Vehicle Year, Make and Model	Renter Make/Model + Renter Vehicle Year	
Repair Facility:	Output	20	Body Shop Name	Repair Facility Name	
	Input	16	Body Shop Phone Number	Telephone Number	
	Output	15	Repair Facility City	City	
	Output	3	Repair Facility State	State	
	Output	7	Repair Facility zip code	Zip Code	
Last Day authorized	Output	10	Date rental is authorized through	CALCULATED	Calculated field. Populated with an Open Ticket only.
Charges to Date:	Output	10	Total Charges	CALCULATED	
Renter Type	Output	10	Claim type	claim type description	
Claims Office:	Output	3	Office Id	external organization abbreviated name	N/A.
Vehicle Condition	Output	15	Type of Loss	loss type description	
Renter Email:	Output	20	Renter's Email	renter email	Will not be able to edit if ticket is Open.

#### 2.1.4 Screen Function Definition

This section includes the definitions for all functions that can be performed within the screen. This includes operations invoked by button clicks, specific shortcut keystrokes, or other actor activity.

##### 2.1.4.1 Skip

When clicked, the USER will be taken out of the use case without changing the current status of the request. Any changes made by clicking Change or Add and keying data in the bottom section will be saved.

##### 2.1.4.2 Process

When clicked, the system will validate the input and accept the changes made to the customer file. The ARMS/Web database will be updated. The use case will then end and the USER will return to the process from which they came.

##### 2.1.4.3 Notebook

When clicked, the USER will be taken to the Note Book section at the bottom of the screen to view all messages for this rental.

##### 2.1.4.4 Set Last Date

When clicked, the system will terminate the rental. The USER will be prompted to enter



a termination date for this rental. This coincides with the use case MA-17-Terminate Rental.

#### **2.1.4.5 Transfer File**

When clicked, the USER will be taken to the Transfer File screen. This screen allows the USER to change the office or adjuster currently assigned to the customer file. The required information in the Extend Rental/Customer File will be passed to the Transfer File screen. Upon completion of the transfer, the USER will then be returned to the next action item or the profiled start page, depending on the screen from which the USER began.

#### **2.1.4.6 Change or Add**

When clicked, the system will refresh the current screen and make all editable fields in the bottom section (outside the gray box area) input capable. The changes on the top of the screen will not be lost.

#### **2.1.4.7 Top of page**

When clicked, the USER will be taken to the top of the current page.

#### **2.1.4.8 View Car Class**

When clicked, the USER will be taken to the View Car Class Use Case. No changes will be lost. Once the USER is finished with this use case, the USER will return to the Extend Rental Use Case.

#### **2.1.4.9 Extend Rental**

When clicked, the system will validate the input and accept the extension AND the changes made to the customer file. The ARMS/Web database will be updated. The use case will then end and the USER will return to the process from which they came.

ARMS Redesign Project – Release 3.0  
Extend Rental

Issue: 1.1  
Issue Date: 10/20/00

---

---

# **Enterprise Rent-A-Car**

## **ARMS Web 3.0 Functional Design Specification Review List - Action Items**

**Version 1.1**

**Last Saved: 9/28/00 2:37 PM**

ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List - Action Items

Issue Date: 10/20/00

## Revision History

Date	Issue	Description	Author
April 20, 2000	0.1	Use Case and Screen Documents created	Mike Slater, Johnny Sands
April 27, 2000	0.1	Linked subdocuments to master document	Cindy Bastean
May 3, 2000	0.2	Removed subdocuments. Formatted according to standards	Cindy Bastean
May 22, 2000	0.3	Incorporated changes per the Cross Team QA.	Michael Slater, Johnny Sands, Deb Ealick, Cindy Bastean
June 8, 2000	0.4	Added data field Information	Cindy Bastean
June 13, 2000	0.5	Added the Zefer look & feel image	Amanda Banta
July 3, 2000	0.6	Updated Screen Fields and Data Fields information	Cindy Bastean
July 14, 2000	0.6	Updated Use Case and Screen Design Sections	Mike Slater, Johnny Sands, Brian Wiengart, Brent Armbruster, Stan Schuchat, Deb Ealick and Cindy Bastean
September 25, 2000	1.1	Changes made based on feedback provided by business leads with respect to the future state of Release 3.0.	Amanda Banta, Aaron Foster, Mike Slater, Tim Weinstock

ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List - Action Items

Issue Date: 10/20/00

## Table of Contents

1.	Review List Action Items Use Case	1
1.1	Application Overview	1
1.2	Brief Description	1
1.3	Use Case Actors	1
1.4	Pre-Conditions	1
1.5	Flow of Events	1
1.5.1	Activity Diagram	2
1.5.2	Basic Flow	3
1.5.3	Alternative Flows	3
1.6	Post-Conditions	3
1.7	Special Requirements	3
1.7.1	Sort Request	3
1.8	Extension Points	4
1.8.1	MA-12-Extend Rental	4
1.8.2	MA-10-Authorize Request	4
1.8.3	Invoicing – BI-01-Handle Unapproved Invoices & BI-02-Pay Approved Invoices & BI-03 Reject an Invoice	4
1.8.4	MA-19-Customer File (Message)	4
2.	Screen Design	2
2.1	Action Items	2
2.1.1	Screen Layout – Action Items	2
2.1.2	Action Items - Summary	3
2.1.3	Screen Function Definition	5

## Review List - Action Items

### 1. Review List Action Items Use Case

#### 1.1 Application Overview

The following is a document used to illustrate the process for how the USER would view and/or select any outstanding action items assigned to them using ARMS/Web 3.0. The intent for this release of the ARMS/Web application is to reach a much wider audience. This application will target a Multi-Vendor, Multi-Segment, and International customer base.

#### 1.2 Brief Description

This use case describes how the USER would view and/or select any outstanding action items assigned to them.

#### 1.3 Use Case Actors

The following actors will interact with this use case.

- **RENTAL ADMINISTRATOR** – The RENTAL ADMINISTRATOR will use the system to review outstanding action items to be completed. This use case refers to a USER in the role of a USER. There are various types of customers that the USER would represent, which include corporate account holders, car dealerships, insurance companies, and others.
- **ARMS** – The ARMS system will receive/send transactions to ARMS/Web based on actions of the USER, retrieving and acting action items.
- **RENTAL CAR COMPANY** – A wide variety of rental car companies will be able to use this system as well. Each company will have the ability to initiate and manage their rentals through the use of this application.

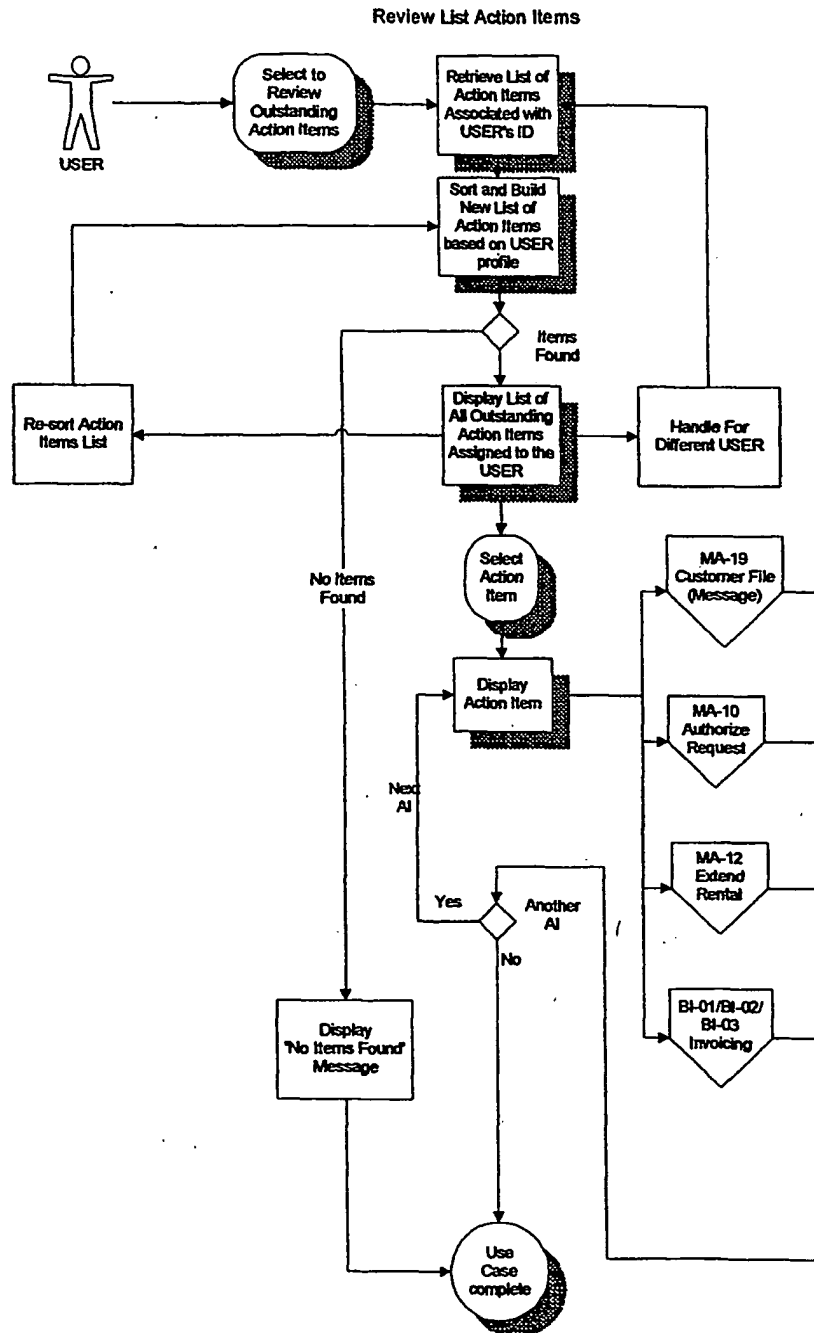
#### 1.4 Pre-Conditions

- The USER must be logged into the ARMS/Web system.
- The USER must have selected to Review a List of Action Items.
- The system must retrieve and confirm the USER ID and access authority.

#### 1.5 Flow of Events

The Flow of Events will include the necessary steps for an USER to review and assign outstanding action items.

## 1.5.1 Activity Diagram



ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List - Action Items

Issue Date: 10/20/00

---

**1.5.2 Basic Flow**

1. The USER selects to review the outstanding action items list.
2. The system retrieves the list of outstanding action items associated with the USER ID.
3. The system sorts and builds the list based on the appropriate USER profile.
4. The system will display a list of all outstanding action items assigned to the USER, which could include:
  - Authorize a Request
  - Extend a Rental
  - Handle Unapproved Invoices/Pay Approved Invoices
  - Send a Message
5. The USER will select an item from the action items list.
6. The system displays the detail appropriate to the action item status.
7. Upon completion of the selected action item, the system will determine the next action item and display until the current list has been completed.
8. This ends the use case.

**1.5.3 Alternative Flows****1.5.3.1 Handle For A Different USER**

Until step 5, the USER may choose to handle requests for another USER. At this time, the USER must select the appropriate USER to handle for. The system will then validate the ID of the alternate USER, and then rebuild the action item list to include all outstanding items associated with the new ID.

**1.5.3.2 Re-sort Action Items List**

After displaying the action item list using the default from the profile, the USER may decide to sort the list based on some other criteria. At any time, the USER may choose to re-sort the action item list (Depending on the USER segment) based on Item Type, Date Received, Renter's Name, Claim Number or Corporate Class Number or Purchase Order Number, Rental Company, and Administrator.

**1.5.3.3 No Items Found**

If there are no Action Items available for the USER work on, the system will display a message indicating that there are no available action items to display.

**1.6 Post-Conditions**

None

**1.7 Special Requirements****1.7.1 Sort Request**

The default sort order has been specified by the USERs profile, which governs the order in which action items have been presented. If invoices have been added to the USER's payment list, a link displays for them to proceed to the 'Payment List'. Alternatively, after the last invoice has been approved, the system automatically proceeds to the 'Payment List' before resuming the outstanding action items. If the USER has been designated with the responsibility of handling the 'Unassigned Requests,' a link at the bottom of the action item list displays.



**1.8 Extension Points**

An extension point indicates a link between this use case and another use case. Extension points associated with the use case are indicated below. Clicking on the extension point will open the related use case.

**1.8.1 MA-12-Extend Rental**

At step 5, the USER must select an action item to perform. At this point, the USER may elect to extend a previously authorized rental. Extensions may be performed due to prolonged body shop delays and other scenarios. Upon completion of the Extend Rental process, the USER should be returned to step 5 of the Basic Flow. The action item that called for the extension should no longer appear in the USER's action item list.

**1.8.2 MA-10-Authorize Request**

At step 5, the USER must select an action item to perform. At this point, the USER may elect to authorize a direct bill request. Upon completion of the authorization, the USER should be returned back to step 5 of the Basic Flow. The request needing authorization should no longer appear in the USER's action item list.

**1.8.3 Invoicing – BI-01-Handle Unapproved Invoices & BI-02-Pay Approved Invoices & BI-03 Reject an Invoice**

At step 5, the USER must select an action item to perform. At this point the USER may elect to pay approved invoices, handle unapproved invoices, or reject an invoice. Upon completion of this process, the USER should be returned back to step 5 of the Basic Flow. The invoices that were processed should no longer appear in the USER's action item list.

**1.8.4 MA-19 – View Customer File (Message)**

At step 5, the USER must select an action item to perform. At this point, the USER may elect to view a message from the rental company. Upon completion of the message, the USER should be returned back to step 5 of the Basic Flow. The message should no longer appear in the USER's action item list.

ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List – Action Items

Issue Date: 10/20/00

## 2. Screen Design

A definition of the screen layout(s), screen data fields, and screen functions that are used to implement the flows identified above. More than one screen may be used to implement support for the use case flow.

### 2.1 Action Items

This screen will allow the USER to pick which functions that he/she may want to change.

#### 2.1.1 Screen Layout – Action Items

(ARMS/Web 2.0)

**Action Items: Welcome back, Fitzgerald, Neil.**

Screen please find the below items that require your attention.

To sort the Action Items, click the column title or use the sorting method (or to sort by date, click "DATE RECEIVED")

ITEM TYPE	DATE RECEIVED	REVIEWER NAME	CLAIM NUMBER	AGENCY
Extension	05-01-00	Bowen, David	234567871	Fitzgerald, Neil
Invoice	05-01-00	Crystal, Emily	234567871	Fitzgerald, Neil

Current User: James G. Conner

ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List - Action Items

Issue Date: 10/20/00

(Insurance User)

Welcome to the  
Automated Rental Management System

create a RESERVATION find a CUSTOMER

Claims Office: 001 Handling for: Yourself

**Action Items: Welcome back, Fitzgerald, Neil.**

Below please find the action items that require your attention.

To sort the Action Items, click the column title of your chosen sorting method  
(ex to sort by date, click "DATE RECEIVED")

TYPE	DATE RECEIVED	RENTER'S NAME	CLAIM NUMBER	RENTAL COMPANY	ADDRESS
Extension	05-01-00	Bowie, David	234587871	HLE Rent-A-Car	Fitzgerald, Neil
Message	05-01-00	Simpson, David	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Invoice	05-01-00	Weber, Andrew	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Returned Vehicle	05-01-00	Crystal, Bill	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Payment List	06-15-00	(5) Invoices		Car Temps	Fitzgerald, Neil
Unsubmitted Items	06-15-00	(7) Action Items		Enterprise Rent-A-Car	Unsubmitted

Over 24 hours old

New Information has come in.

[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

(Fleet User)

Welcome to the  
Automated Rental Management System

create a RESERVATION find a CUSTOMER

Office: 001 Handling for: Yourself

**Action Items: Welcome back, Fitzgerald, Neil.**

Below please find the action items that require your attention.

To sort the Action Items, click the column title of your chosen sorting method  
(ex to sort by date, click "DATE RECEIVED")

TYPE	DATE RECEIVED	RENTER'S NAME	CLAIM NUMBER	RENTAL COMPANY	ADDRESS
Extension	05-01-00	Bowie, David	234587871	HLE Rent-A-Car	Fitzgerald, Neil
Message	05-01-00	Simpson, David	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Invoice	05-01-00	Weber, Andrew	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Returned Vehicle	05-01-00	Crystal, Bill	754589877	Enterprise Rent-A-Car	Fitzgerald, Neil
Payment List	06-15-00	(5) Invoices		Car Temps	Fitzgerald, Neil
Unsubmitted Items	06-15-00	(7) Action Items		Enterprise Rent-A-Car	Unsubmitted

Over 24 hours old

New Information has come in.

[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

Confidential

Enterprise Rent-A-Car © 2000

3

Y:\APPS\ARMSWeb\Application\ARMSWEB\VT\ddb\Functional Specs\MA-14 Review List Action Items\Functional Specs\MA-14-FDS-ARMSWeb3.0v1.1.doc

ARMS Redesign Project – Release 3.0

Issue: v1.1

Review List - Action Items

Issue Date: 10/20/00

(Dealership User)

Welcome to the  
Automated Rental Management System

create a RESERVATION find a CUSTOMER

Office:  Handling for:

**Action Items: Welcome back, Fitzgerald, Neil.**

Below please find the action items that require your attention.

To sort the Action Items, click the column title of your chosen sorting method  
(ec to sort by date, click "DATE RECEIVED")

TYPE	DATE RECEIVED	SERVER NAME	PERSONAL CLASS NUMBER	RENTAL COMPANY	ADMINISTRATOR
Direct Bill Request	04-27-00	Banko, Tim	734589671	Enterprise Rent-A-Car	Fitzgerald, Neil
Extension	05-01-00	Bowie, David	234587871	HLE Rent-A-Car	Fitzgerald, Neil
Message	05-01-00	Simmons, Homer	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Invoice	05-01-00	Weber, Andrew	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Returned Invoice	05-01-00	Crystal, Billy	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Payment List	06-15-00	(5) Invoices		Car Temps	Fitzgerald, Neil
Unassigned Items	06-15-00	(2) Action Items		Enterprise Rent-A-Car	Unassigned

Over 24 hours old

New Information has come in.

[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

(Corporate User)

Welcome to the  
Automated Rental Management System

create a RESERVATION find a CUSTOMER

Office:  Handling for:

**Action Items: Welcome back, Fitzgerald, Neil.**

Below please find the action items that require your attention.

To sort the Action Items, click the column title of your chosen sorting method  
(ec to sort by date, click "DATE RECEIVED")

TYPE	DATE RECEIVED	SERVER NAME	CORPORATE CLASS NUMBER	RENTAL COMPANY	ADMINISTRATOR
Direct Bill Request	04-27-00	Banko, Tim	734589671	Enterprise Rent-A-Car	Fitzgerald, Neil
Extension	05-01-00	Bowie, David	234587871	HLE Rent-A-Car	Fitzgerald, Neil
Message	05-01-00	Simmons, Homer	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Invoice	05-01-00	Weber, Andrew	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Returned Invoice	05-01-00	Crystal, Billy	754589677	Enterprise Rent-A-Car	Fitzgerald, Neil
Payment List	06-15-00	(5) Invoices		Car Temps	Fitzgerald, Neil
Unassigned Items	06-15-00	(2) Action Items		Enterprise Rent-A-Car	Unassigned

Over 24 hours old

New Information has come in.

[Contact Us](#) | [Terms & Conditions](#) | [Log Off](#)

ARMS Redesign Project – Release 3.0  
Review List - Action Items

Issue: v1.1  
Issue Date: 10/20/00

### 2.1.2 Action Items - Summary

Screen Label	Type	Size	Screen Field Name	Data Field	Screen Specific Rule
Date Received	Output	0	Date Received	action item assigned date	N/A.
Type	Output	15	Action Item Type	action item type description	N/A.
USER	Output	0	USER's Name	First Name + Last Name	N/A.
Handling For:	List Box	30	Handling for USER's Name	First Name + last Name	N/A.
Welcome Back	Output	30	User's Name	Last Name + First Name	N/A.
Claim Number Purchase Order Number Corporate Class Number	Output	0	Claim Number Purchase Order Number Corporate Class Number	Insurance Claim Number, PO#, CC#	N/A.
Renter's Name	Output	30	Renter's Name	First Name + Last Name	N/A.
Claims Office:	List Box	3	Office	external organization abbreviated name	

### 2.1.3 Screen Function Definition

This section includes the definitions for all functions that can be performed within the screen. This includes operations invoked by button clicks, specific shortcut keystrokes, or other actor activity.

#### 2.1.3.1 Renter's Name

When clicked on a specific hyperlink under the "Renter's Name" heading, the USER will go into the details of that particular action item and will begin any of the following use cases:

- MA-12-Extend Rental
- MA-10-Authorize Request
- Invoicing – BI-01-Handle Unapproved Invoices & BI-02-Pay Approved Invoices & BI-03 Reject an Invoice
- MA-19-Customer File (Message)

---

# **Enterprise Rent-A-Car**

## **ARMS Web 3.0 Functional Design Specification Assign a Request**

**Version 1.1**

**Last Saved: 9/28/00 4:07 PM**

ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

## Revision History

Date	Issue	Description	Author
April 7, 2000	0.1	Initial Draft	Keith Baker
April 11, 2000	0.2	Added fields from Databases	Cindy Bastean / Debi Ealick
April 20, 2000	0.2	Merged subdocuments	Cindy Bastean
April 27, 2000	0.3	Added new screen layouts	Deb Ealick
April 27, 2000	0.4	Removed Application Operations and Data Fields per new Functional Spec Reporting Process Implementation	Cindy Bastean
May 10, 2000	0.5	Included changes from Cross-Team QA	Cindy Bastean
May 15, 2000	0.6	Added screen change to document according to cross-team review	Debi Ealick
June 8, 2000	0.6	Added Data Field Information	Cindy Bastean
July 3, 2000	0.7	Change screen field / data field information	Cindy Bastean
July 13, 2000	0.7	Updated Use Case and Screen Design sections for sign-off	Mike Slater, Brian Weingart, Stanley Schuchat, Deb Ealick, Brent Armbruster, Johnny Sands and Cindy Bastean
August 28, 2000	1.0	Updated Use Case. Changes made based on feedback provided by management reviewers.	Mike Slater
September 26, 2000	1.1	Changes made based on feedback provided by business leads with respect to the future state of Release 3.0.	Michael Slater, Aaron Foster, Amanda Banta, Tim Weinstock

ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

---



ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

## Table of Contents

1.	Assign a Request Use Case	1
1.1	Application Overview	1
1.2	Brief Description	1
1.3	Use Case Actors	1
1.4	Pre-Conditions	1
1.5	Flow of Events	1
1.5.1	Activity Diagram	2
1.5.2	Basic Flow	3
1.5.3	Alternative Flows	3
1.6	Post-Conditions	3
1.7	Special Requirements	3
1.8	Extension Points	3
1.8.1	MA-04 Send Message	3
1.8.2	MA-10 Authorize a Request	3
2.	Screen Design	4
2.1	Action Items – Unassigned	4
2.1.1	Screen Layout – Action Items – Unassigned (ARMS Web 2.0)	4
2.1.2	Action Items - Unassigned	7
	Screen Function Definition	7

## Assign a Request

### 1. Assign a Request Use Case

#### 1.1 Application Overview

The following is a document used to illustrate the process for assigning the unassigned authorization requests to the appropriate user. The assignments will be made using the ARMS Web 3.0 system. The intent for this release of the ARMS Web application is to reach a much wider audience. This application will target a Multi-Vendor, Multi-Segment, and International customer base.

#### 1.2 Brief Description

This use case describes the process of how a USER will review unassigned authorization request and assign them to a USER for further handling.

#### 1.3 Use Case Actors

The following actors will interact with this use case:

- **RENTAL ADMINISTRATOR** – RENTAL ADMINISTRATOR will use the system to assign the unassigned authorization requests. This use case refers to a USER in the role of a rental administrator. There are various types of customers that the rental administrator would represent, which include corporate account holders, car dealerships, insurance companies, and others.
- **ARMS** – The ARMS system will receive/send transactions to ARMS Web to manage each phase of the rental process.
- **RENTAL CAR COMPANY** – A wide variety of rental car companies will be able to use this system as well. Each company will have the ability to initiate and manage their rentals through the use of this application.

#### 1.4 Pre-Conditions

- The USER must be signed-on to the ARMS Web system.
- The USER should be authorized to assign a request.
- If there are unassigned requests present, the USER has selected the link from the Review List Action Items Use Case to enter this use case.

#### 1.5 Flow of Events

The Flow of Events will include the necessary steps to make changes and updates to "Assign an Action Item".

ARMS Redesign Project - Release 1.0

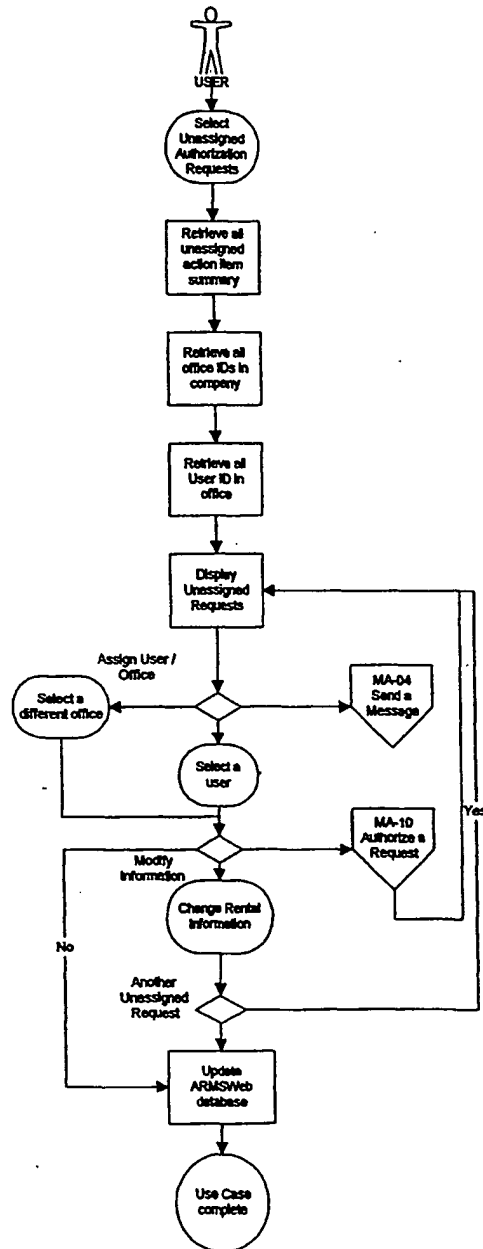
Issue: v1.1

Assign a Request

Issue Date: 10/20/00

### 1.5.1 Activity Diagram

Assign An Action Item Diagram



ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a RequestIssue Date: 10/20/00

---

**1.5.2 Basic Flow**

1. The USER selects the unassigned authorizations link.
2. The system retrieves all unassigned request summaries.
3. The system retrieves all OFFICE IDs within ARMS Web.
4. The system retrieves all USER IDs within the OFFICE.
5. The system displays the unassigned authorization summaries with the offices and users.
6. The USER selects a user to assign to the request.
7. The system will update the ARMS Web database.
8. This ends the use case.

**1.5.3 Alternative Flows****1.5.3.1 Cancel Use Case**

The USER should be capable of leaving the use case at any point prior to assigning the of the reservation information.

**1.5.3.2 Modify a Request**

Before step 6 of the basic flow, the USER should be able to make changes to the authorization.

**1.5.3.3 Select a different office**

Before step 6 of the basic flow, the USER should be able to select a different office for this authorization request. If a different office has been selected, the user cannot assign the file to a new user. The new office must now assign the file.

**1.6 Post-Conditions**

If the use case is successful, the system will change the request type from an unassigned authorization request to direct bill. If the user has authority to authorize this request, the system will change the request to Authorized status and assign the adjuster picked in Step 5 of the basic flow.

If the use case is unsuccessful, the system state will remain unchanged.

**1.7 Special Requirements**

None

**1.8 Extension Points****1.8.1 MA-04 Send Message**

The Send Message function will be used to allow the user to capture messages and diary notes associated with a rental reservation/authorization. The USER can elect to have the message sent to the rental branch location responsible for the reservation/authorization. The USER may also send a message without assigning the file to a user/office. All MESSAGES and DIARY NOTES captured must be related to a specific reservation/authorization.

**1.8.2 MA-10 Authorize a Request**

The USER may decide to enter into the full detail screen of the unassigned request, which would invoke the Authorize a Request use case.

ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

## 2. Screen Design

A definition of the screen layout(s), screen data fields, and screen functions that are used to implement the flows identified above. More than one screen may be used to implement support for the use case flow.

### 2.1 Action Items – Unassigned

This screen will allow the USER to assign action items to an office or USER. The USER may also cancel an item or change specified information in the Customer File through this screen.

#### 2.1.1 Screen Layout – Action Items – Unassigned (ARMS Web 2.0)

**Automated Rental Management System**

CLAIMS OFFICE: 001 Handling for: Yourself

You just approved an invoice for Crystal, Billy  
Total Amount \$536.13

**Action Items:  
UNASSIGNED**

---

**Weber, Andrew**  
28445 Main Ave  
Chicago, IL 60622  
555-555-1212

**DIRECT BILL REQUEST**  
Claim Number: 754589877  
Vehicle Condition: Select a Loss Type  
Claim Type: Select a Claim Type  
Date of Loss: January 1 2000  
Note to Enterprise:

Assign to Office: 001  
Assign Adjuster: Unassigned  
Cancel this item

---

**Smith, Joe**  
28445 Main Ave  
Chicago, IL 60622  
555-555-1212

**DIRECT BILL REQUEST**  
Claim Number: 754589877  
Vehicle Condition: Select a Loss Type  
Claim Type: Select a Claim Type  
Date of Loss: January 1 2000  
Note to Enterprise:

Assign to Office: 001  
Assign Adjuster: Unassigned  
Cancel this item

[Contact Us](#) | [Terms & Conditions](#)

ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

(Insurance User)

Welcome to the  
Automated Rental Management System

CREATE A RESERVATION find a CUSTOMER

Claims Office: 001 Handling for: Yourself

You just approved an invoice for Crystal, Billy - Total Amount \$536.13

**Action Items:**  
UNASSIGNED

<b>Weber, Andrew</b> 28445 Main Ave Chicago, IL 60622 555-555-1212	<b>DIRECT BILL REQUEST</b> Claim Number: 754589877 Vehicle Condition: Select a Loss Type Claim Type: Select a Claim Type Date of Loss: January 1 2000 Note to Rental Company:	1 Assign to Office: 001 2 Assign Adjuster: Unassigned 3 Cancel this item
<b>Rental Location:</b> Enterprise Edgewater Branch 5400 N. Ashland Chicago, IL 60622 773-334-5400		
<b>Smith, Joe</b> 28445 Main Ave Chicago, IL 60622 555-555-1212	<b>DIRECT BILL REQUEST</b> Claim Number: 754589877 Vehicle Condition: Select a Loss Type Claim Type: Select a Claim Type Date of Loss: January 1 2000 Note to Rental Company:	1 Assign to Office: 001 2 Assign Adjuster: Unassigned 3 Cancel this item
<b>Rental Location:</b> Enterprise Edgewater Branch 5400 N. Ashland Chicago, IL 60622 773-334-5400		

Contact Us | Terms & Conditions | Log Off

(Insurance Fleet)

Welcome to the  
Automated Rental Management System

CREATE A RESERVATION find a CUSTOMER

Office: 001 Handling for: Yourself

You just approved an invoice for Crystal, Billy - Total Amount \$536.13

**Action Items:**  
UNASSIGNED

<b>Weber, Andrew</b> 28445 Main Ave Chicago, IL 60622 555-555-1212	<b>DIRECT BILL REQUEST</b> Claim Number: 754589877 Protection Coverage: Claim Type: Select a Claim Type Date of Loss: January 1 2000 Note to Rental Company:	1 Assign to Office: 001 2 Assign Administrator: Unassigned 3 Cancel this item
<b>Rental Location:</b> Enterprise Edgewater Branch 5400 N. Ashland Chicago, IL 60622 773-334-5400		
<b>Smith, Joe</b> 28445 Main Ave Chicago, IL 60622 555-555-1212	<b>DIRECT BILL REQUEST</b> Claim Number: 754589877 Protection Coverage: Claim Type: Select a Claim Type Date of Loss: January 1 2000 Note to Rental Company:	1 Assign to Office: 001 2 Assign Administrator: Unassigned 3 Cancel this item
<b>Rental Location:</b> Enterprise Edgewater Branch 5400 N. Ashland Chicago, IL 60622 773-334-5400		

Contact Us | Terms & Conditions | Log Off

ARMS Redesign Project - Release 1.0

Issue: v1.1

Assign a Request

Issue Date: 10/20/00

(Dealership User)

Welcome to the  
Automated Rental Management System

CREATED A RESERVATION first A CUSTOMER

Office: 001 Handling for: Yourself

You just approved an invoice for Crystal, Billy - Total Amount \$536.13

**Action Items:**  
UNASSIGNED

**Weber, Andrew**  
28445 Main Ave  
Chicago, IL 60622  
655-655-1212

DIRECT BILL REQUEST  
Purchase Order No. 754589877  
Bill Type:

Rental Location:  
Enterprise Edgewater Branch Date of Loss: January 1 2000  
6400 N. Ashland  
Chicago, IL 60622 Note to Rental Company:   
773-334-6400

1 Assign to Office: 001  
2 Assign Administrator: Unassigned  
3 Cancel this item

**Smith, Joe**  
28445 Main Ave  
Chicago, IL 60622  
655-655-1212

DIRECT BILL REQUEST  
Purchase Order No. 754589877  
Bill Type:

Rental Location:  
Enterprise Edgewater Branch Date of Loss: January 1 2000  
6400 N. Ashland  
Chicago, IL 60622 Note to Rental Company:   
773-334-6400

1 Assign to Office: 001  
2 Assign Administrator: Unassigned  
3 Cancel this item

Contact Us | Terms & Conditions | Log Off

(Corporate User)

Welcome to the  
Automated Rental Management System

CREATED A RESERVATION first A CUSTOMER

Office: 001 Handling for: Yourself

You just approved an invoice for Crystal, Billy - Total Amount \$536.13

**Action Items:**  
UNASSIGNED

**Weber, Andrew**  
28445 Main Ave  
Chicago, IL 60622  
655-655-1212

DIRECT BILL REQUEST  
Corporate Class No. 754589877  
Note to Rental Company:

Rental Location:  
Enterprise Edgewater Branch  
6400 N. Ashland  
Chicago, IL 60622  
773-334-6400

1 Assign to Office: 001  
2 Assign Administrator: Unassigned  
3 Cancel this item

**Smith, Joe**  
28445 Main Ave  
Chicago, IL 60622  
655-655-1212

DIRECT BILL REQUEST  
Corporate Class No. 754589877  
Note to Rental Company:

Rental Location:  
Enterprise Edgewater Branch  
6400 N. Ashland  
Chicago, IL 60622  
773-334-6400

1 Assign to Office: 001  
2 Assign Administrator: Unassigned  
3 Cancel this item

Contact Us | Terms & Conditions | Log Off

ARMS Redesign Project - Release 1.0  
Assign a Request

Issue: vl.1  
Issue Date: 10/20/00

### 2.1.2 Action Items - Unassigned

Screen Label	Type	Size	Screen Field Name	Data Field Name	Screen Specific Rule
Claims Office:	Output	3	Office Id	external organization abbreviated name	N/A.
Handling For:	Output	30	Handling for Adjuster's Name	First Name + Last Name	N/A.
	Output	30	Renter's Name	First Name + Last Name	This should be a link. The USER should be able to get to the authorize page from this screen field
	Output	30	Renter's Address	Address Line	
	Output	10	Renter's City	City	
	Output	3	Renter's State	State	
	Output	10	Renter's Zip Code	Zip Code	
	Output	16	Renter's Home Phone	Renters Night Phone + Renters Night Phone Extension	If these fields are populated, add a label to the screen to differentiate between Home Phone and Work Phone
	Output	16	Renter's Work Phone	Day Phone + Renters Day Phone Extension	If these fields are populated, add a label to the screen to differentiate between Home Phone and Work Phone
Claim Number Purchase Order Number Corporate Class Number	Input	30	Claim Number Purchase Order Number Corporate Class Number	Insurance Claim Number, PO#, CC#	N/A.
Vehicle Condition	List Box	15	Loss Type	loss type description	
Claim Type Bill Type	List Box	15	Claim Type Bill Type	Rental type description	N/A.
Date of Loss:	Input	10	Date of Loss	Date Of Loss	N/A.
Note to Enterprise	Input	30	Message Text	NOTE	N/A.
Assign to office:	List Box	5	Office Id	external organization abbreviated name	
Assign adjuster:	List Box	30	Adjuster Name:	First Name + Last Name	Lists only those adjusters the USER has authority to assign

#### Screen Function Definition

This section includes the definitions for all functions that can be performed within the screen. This includes operations invoked by button clicks, specific shortcut keystrokes, or other actor activity.

##### 2.1.2.1 <<Previous

When clicked, the USER will be taken back to the previous screen.

##### 2.1.2.2 Process

When clicked, the USER will be taken to the next item in the action item list or a detail of the completed action items. This button ends the use case



ARMS Redesign Project - Release 1.0  
Assign a Request

Issue: v1.1  
Issue Date: 10/20/00

---

### 2.1.2.3 Cancel

When clicked, the USER will be allowed to cancel the authorization. If this occurs, the rental becomes unauthorized and the rental is no longer responsibility of the company.

---

# **Enterprise Rent-A-Car**

## **ARMS/Web 3.0 Functional Design Specification View Car Class**

**Version 1.3**

**Last Saved: 9/27/00 9:33 AM**

ARMS Redesign Project – Release 3.0

Issue: 1.3

View Car Class

Issue Date: 10/20/00

## Revision History

Date	Issue	Description	Author
2000-04-07	0.1	Initial draft published to design team for review and comment.	Brent Armbruster, Sean O'Donnell
2000-04-11	0.2	Removed subdocuments. Updated data fields listing to reflect format changes	Cindy Bastean
2000-05-01	0.3	Added screen fields to "Car Class Summary" and "Car Detail Screen" screen field tables	Cindy Bastean
2000-05-08	0.4	Included additional changes, comments and feedback out of the initial review sessions.	Sean O'Donnell
2000-05-30	0.5	Added data field information from DATA_FIELDS	Cindy Bastean
2000-09-27	1.3	Changes made based on feedback provided by business leads with respect to the future state of Release 3.0.	Amanda Banta, Aaron Foster, Mike Slater, Tim Weinstock

ARMS Redesign Project – Release 3.0  
View Car Class

Issue: 1.3  
Issue Date: 10/20/00

---

## Table of Contents

1.	View Car Class Use Case	1
1.1	Application Overview	1
1.2	Brief Description	1
1.3	Use Case Actors	1
1.4	Pre-Conditions	Error! Bookmark not defined.
1.5	Flow of Events	1
1.5.1	Activity Diagram	2
1.5.2	Basic Flow	3
1.5.3	Alternative Flows	3
1.6	Post-Conditions	3
1.7	Special Requirements	4
1.7.1	Modify Car Class Selection Results	4
1.8	Extension Points	4
2.	Screen Design	5
2.1	Car Class Detail Screen	5
2.1.1	Screen Layout	5
2.1.2	Car Class Details	6
2.1.3	Screen Function Definition	7
3.	Questions and Answers	8

## View Car Class

### 1. View Car Class Use Case

#### 1.1 Application Overview

The following is a document used to illustrate the process for how the USER would view examples of automobiles that are part of each rental company car class using ARMS/Web 3.0. The intent for this release of the ARMS/Web application is to reach a much wider audience. This application will target a Multi-Vendor, Multi-Segment, and International customer base.

#### 1.2 Brief Description

This use case will allow the USER to view examples of automobiles that are part of each rental company car class. The USER will have the ability to select a car class and have the rate for the car class apply to the reservation/authorization.

#### 1.3 Use Case Actors

The following actors will interact with this use case:

- **RENTAL ADMINISTRATOR** – The RENTAL ADMINISTRATOR will use the system to view and/or select the car class that will apply to a reservation. This use case refers to a USER in the role of a USER. There are various types of customers that the USER would represent, which include corporate account holders, car dealerships, insurance companies, and others.
- **ARMS** – The ARMS system will receive/send transactions to ARMS/Web to retrieving information regarding the automobiles.
- **RENTAL CAR COMPANY** – A wide variety of rental car companies will be able to use this system as well. Each company will have the ability to initiate and manage their rentals through the use of this application.

#### 1.4 Pre-Conditions

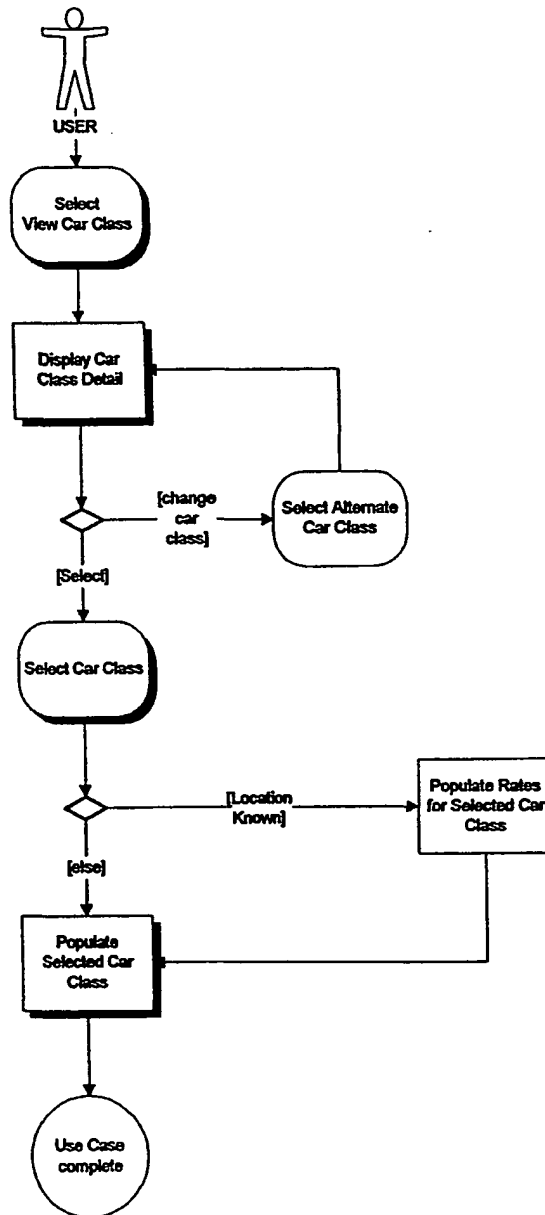
- The USER must be signed-on to the ARMS/Web system.
- The USER must have a reservation or open ticket selected.

#### 1.5 Flow of Events

The Flow of Events will include the necessary steps to view and/or select the car class to apply to a rental reservation.

## 1.5.1 Activity Diagram

View Car Class Activity Diagram



### 1.5.2 Basic Flow

The Basic Flow of the View Car Class use case includes all of the required steps to view and/or select a car class for a rental reservation. If a car class is selected, it will be used to populate rate information on a rental authorization.

1. The USER will select View Car Class from the active reservation or open ticket.
2. The system will display a car class detail screen. If the USER had previously selected a car class (for example, on the Create Reservation screen), the car class selected will be displayed. If no car class has been selected, the system will display the Standard car class.
3. The USER will select the car class to apply to the reservation or open ticket.
4. The system will return the USER to the active reservation or open ticket and populate car class information based on the car class selected.
5. This ends this use case.

### 1.5.3 Alternative Flows

#### 1.5.3.1 Select Alternate Car Class

From Step 2 of the Basic Flow, the USER will have the ability to view an alternate car class. The car classes that will be available to view include:

- Economy
- Compact
- Intermediate
- Standard
- Full Size
- Premium

If the USER selects an alternate car class, the system will refresh and present the details of the new car class.

#### 1.5.3.2 Populate Car Class Rates

If a rental branch location has already been selected prior to entering this use case, the selection of a car class will populate the rates that apply to the selected car class on the active reservation or open ticket. This alternate flow returns the USER to Step 4 of the Basic Flow.

### 1.6 Post-Conditions

- If successful, the selected Car Class will be returned to the active reservation or open ticket.
- If unsuccessful, the system state is unchanged.

ARMS Redesign Project – Release 3.0

Issue: 1.3

View Car Class

Issue Date: 10/20/00

---

**1.7 Special Requirements**

The additional requirements of the business use case are included here. These are requirements not covered by the flow as they have been described in the sections above.

**1.7.1 Modify Car Class Selection Results**

The USER may change the results of this use case as part of the active reservation or open ticket.

**1.8 Extension Points**

None.



ARMS Redesign Project – Release 3.0  
View Car Class

Issue: 1.3  
Issue Date: 10/20/00

## 2. Screen Design

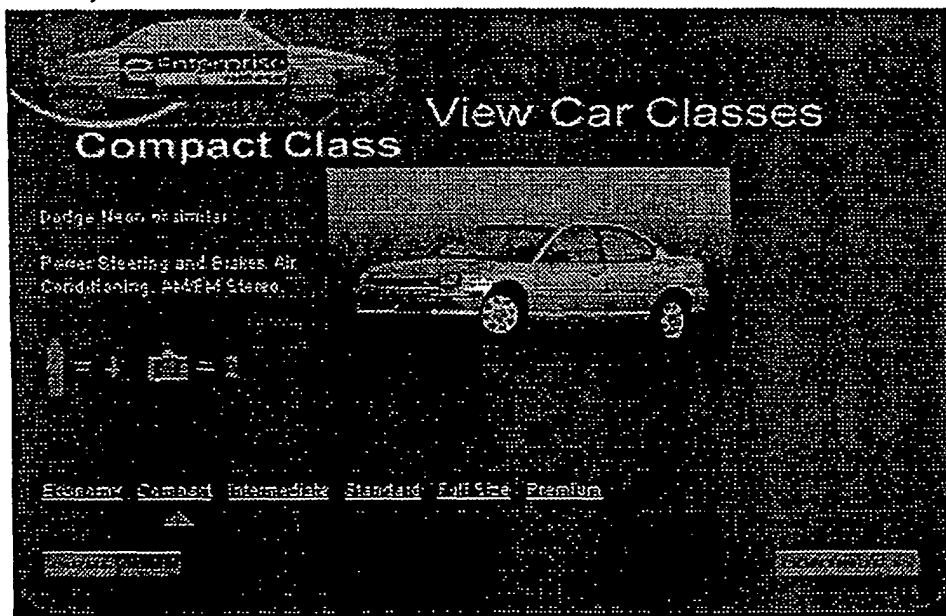
A definition of the screen layout(s), screen data fields, and screen functions that are used to implement the flows identified above. More than one screen may be used to implement support for the use case flow.

### 2.1 Car Class Detail Screen

This screen will allow the USER to view detailed information about the rental company's car classes. The USER will also have the ability to select a car class to apply to a rental reservation / authorization.

#### 2.1.1 Screen Layout

(ARMS/Web 2.0)



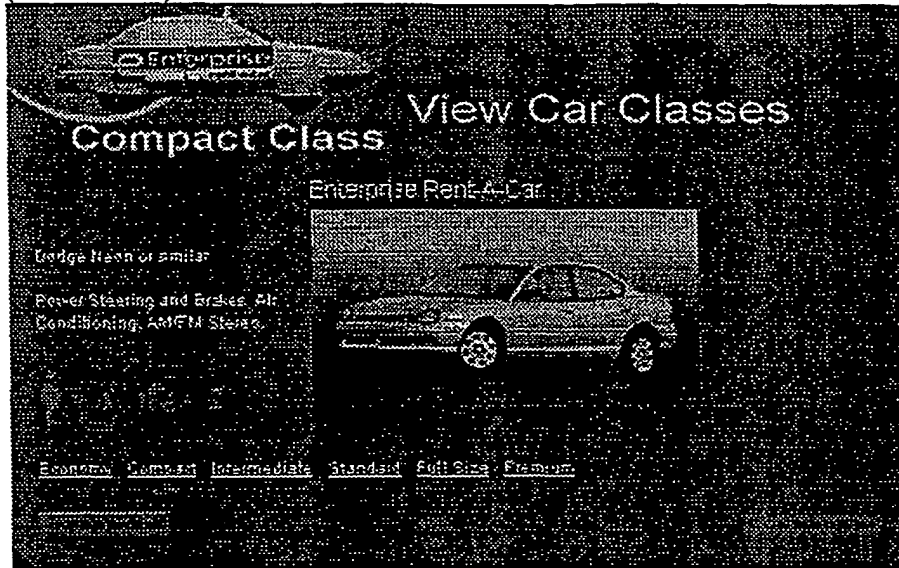
ARMS Redesign Project – Release 3.0

Issue: 1.3

View Car Class

Issue Date: 10/20/00

(ARMS/Web 3.0)



### 2.1.2 Car Class Details

Screen Label	Type	Length	Screen/field Name	Data field	Screen/Spec/field Rule
	Output	20	Car Class Name		This should be the name of the currently selected car class
	Output	40	Rental Company Name		
(Person Image)	Output	2	Car Class Person Capacity		This should provide the average person capacity of the selected car class.
(Luggage Image)	Output	2	Car Class Luggage Capacity		This should provide the average luggage capacity of the selected car class
	Hidden	255	Car Class Image Source		This should provide a picture of an example car within the selected car class.
	Output	120	Car Class Detail Description		This should provide a description of the selected car class.
Economy	Output		Economy Car Class		This should be a hyperlink to the Economy car class detail.
Compact	Output		Compact Car Class		This should be a hyperlink to the Compact car class detail.
Intermediate	Output		Intermediate Car Class		This should be a hyperlink to the Intermediate car class detail.
Standard	Output		Standard Car Class		This should be a hyperlink to the Standard car class detail.
Full Size	Output		Full Size Car Class		This should be a hyperlink to the Full Size car class detail.
Premium	Output		Premium Car Class		This should be a hyperlink to the Premium car class detail.

### **2.1.3 Screen Function Definition**

This section includes the definitions for all functions that can be performed within the screen. This includes operations invoked by button clicks, specific shortcut keystrokes, or other actor activity.

#### **2.1.3.1 Select This Car Class**

The Continue screen function will allow the USER to select the car class to apply to a reservation.

2.1.3.1.1 The Continue screen function is invoked through either a button click or through an Enter keystroke.

#### **2.1.3.2 Previous**

The Previous screen function allows the USER to return to the previous screen.

2.1.3.2.1 The Previous screen function is invoked through a button click.

ARMS Redesign Project – Release 3.0  
View Car Class

Issue: 1.3  
Issue Date: 10/20/00

---

### 3. Questions and Answers

None.